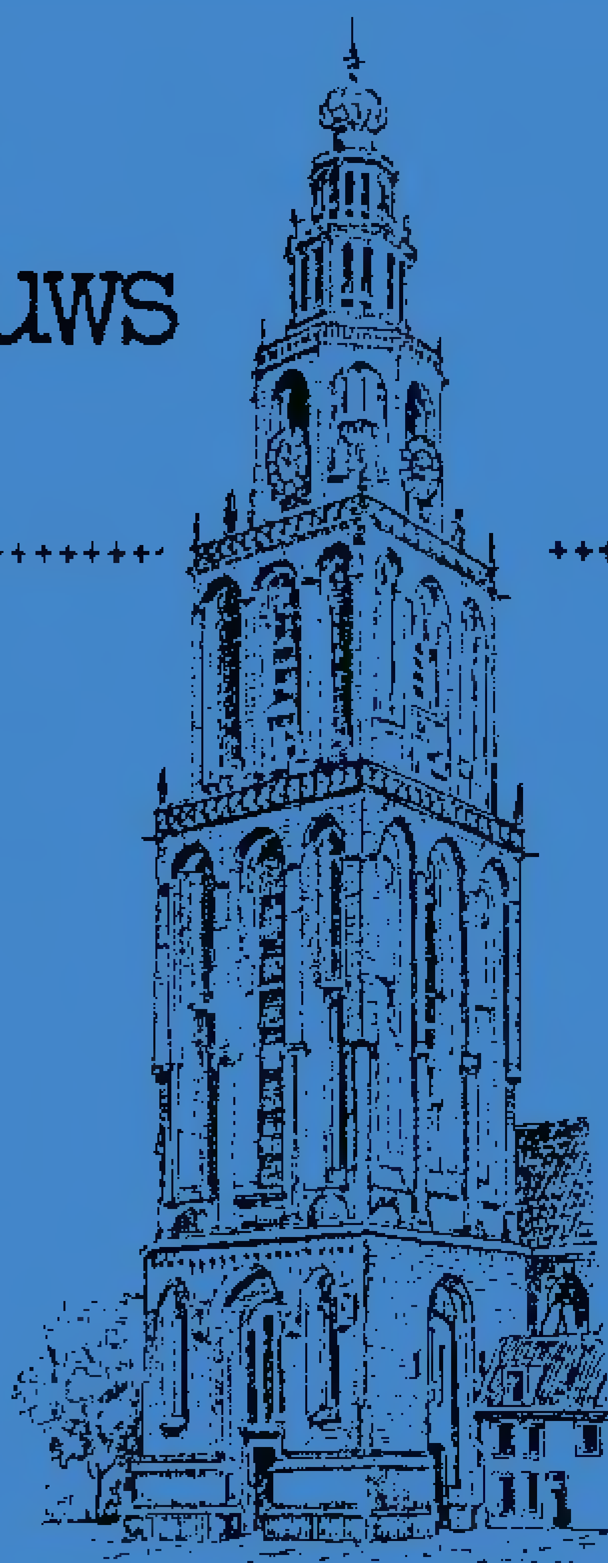
jrg: ●●●
nr: ●

maart 1984

+++

ACORN Nieuws



+++ ++++++

Sound-board
Big benny

enz. etc. uzw.

+++

ACORN NIEUWS Uitg.Fed.Acorn Computerclubs Ned/B.

Verschijnt 6-8 x p/jr.

DE CURSOR nr1.
Spandoek(zie dit nummer). Invaders sneller. Intelligente ATOM schrijfmacine, een machinetaal programma om de computer zonder tekstverwerker als type-machine te gebruiken. Een persbericht over de nieuwe mos 6502 processor-familie. Teken-programma van een kerstboom. Artikel uit MINI-MICROCOMPUTER over de inktstraal-printer. Lissajous figuren en een programma met joy-sticks.

ACORNTJESBROOD december 83.
Pipro2, programma's voor printbezitters. Josbox verbeteringen. (zie dit nummer). Ruisvrij plotten. Sweet16 deel 2. Joystick aanpassingen voor 5 spelletjes. Text statement (zie dit nummer). Doolhof generator. De programmeertal Forth. Kalender-programma. CHAnge string statement om basic-programma's te veranderen.

ACORNTJESBROOD februari 84.
Uitleg over de symbolische assembler. Assemblerversie van het case-statement. Digitiser printontwerp. lissajous figuren. REAL TIME KLOK met de MM58174. Pipro 3. Joystick-uitleesroutine. LIFE programma met uitleg. Toepassingen van morse telex en fax. vervolgverhaal over FORTH.

NEWSLETTER nr1.
Joystickaanpassingen voor twee joysticks. Atom plotting power, voorbeelden om in plot-programma'stegebruiken. Beveiliging tegen te hoge spanningen met de MC3423. Hoe de DOS catalogus in elkaar zit. Een sorteer-programma. DISCASS saved een diskette naar een bandje. Een nieuwe error-handler voor de ATOM. Een 57 files catalogus voor de DOS.

DE STACKER januari 84.
Floatingpoint wijzigingen. Instructief verhaal over het laden en saven. Voorbeeld P-Charme programma van de ackermann-functie. Kalender-programma. Programma voorher maken van een ASCII-tabel.

DE STACKER februari 84.
Opsomming van de inhoud van bandjes archief. De Kat in de Zak multoplexer. Spelletje Memory. Frequentie tellertje. P-charme multidimensionale arrays.

Tot zover de binnen gekomen regio-bladen. Het is een gewoonte dat de regio-redacteuren elkaar hun blad toesturen. En ik moet mij al sterk vergissen als U daar niet in mag neuzen.

voorzitter: Th. v. Kempen
Het Puyven 71
5672 RB Nuenen
tel: 040-836210

secretaris: nog even wachten en deze functie wordt vervuld.
tijdelijk correspondentieadres: A. Jongeling
Klokkengietershoeve 3
7326 SB Apeldoorn
tel: 055-417314

penningmeester: G. Visser
Haringvliet 391
8032 HL Zwolle
tel: 038-546561

postgiro: 5244293 bankgiro: 93.32.87.283
Beide tnv penningmeester Acorn Computerclub te Zwolle.

redactie-adres Açorn Nieuws: H. Reinders
Leeuwarderstraat 8
9718 HX Groningen
telf: 050 - 125458

Alweer het eerste nummer van uw lijfblad door de nieuwe redactiegroep. Weer een dikke, is beloofd.

Contributie 1984.

Eigenlijk is het niet mogelijk, maar het komt toch voor; enkele leden hebben nog steeds geen contributie betaald. Hier een laatste verzoek aan hen.

Voor de leden die de contributie voor 1984 nog niet overgemaakt hebben is dan ook de volgende oproep:

Degenen die om wat voor reden dan ook niet in het bezit zijn van de acceptgirokaart (bv omdat zij later lid zijn geworden) worden verzocht de contributie met een gewone overschrijving te voldoen op een van beide bovenstaande gironummers. Vermeldt u er dan wel even bij dat het gaat om de contributie van 1984 en van welke regio u lid bent? Heeft u nog een acceptgiro dan kunt u die natuurlijk gebruiken.

Wanneer bij het uitkomen van de tweede Acorn Nieuws van 1984 nog geen betaling van iemand ontvangen is, wordt aangenomen dat hij/zij geen lid meer is. Dat uitkomen van nr. 2 laat niet al te lang op zich wachten. Dus om te zorgen dat het blad wat u nu leest niet de laatste is, haast u dan naar de briefenbus met uw overschrijving. Dan krijgt u de andere 7 (dikke) nummers ook nog.

Archiefdiensten.
Voor bandjes, datasheets, EPROM's en alle andere vormen van dienstverlening kunt u terecht bij de regionale archiefdiensten. Mocht in uw regio iets ontbreken, dan kan het regionaal bestuur u verder op weg helpen.
De landelijke archiefdiensten geven regelmatig een overzicht uit van wat er in die archieven aanwezig is. De regionale archiefdiensten kunnen daaruit dan een keus maken van wat zij in het archief willen hebben.
Een overzicht van de besturen en de archiefdiensten vindt u elders in dit nummer.
Let u vooral op het DRUKWERKARCHIEF, dat is namelijk onlangs overgegaan in andere handen.

De "clubwinkel".

Geheugenkaart; 16 kByte extra in de Atom	fl. 35,--
Schakelkaart; meerdere EPROM's op Axxx (incl. 74LS133)	fl. 45,--
Programmerkaart; zelf programmeren van EPROM's	fl. 17,--
Herdruk Acorn Nieuws 1982; 97 pag. wetenswaardigheden	fl. 6,--
Jaargang 1983; totaal ruim 450 pag. compleet leesvoer	fl. 30,--

Bovenstaande artikelen kunt u bestellen door het desbetreffende bedrag over te maken op een van de gironummers. U kunt natuurlijk ook de bestelling via de regionale penningmeesters laten lopen, zodat u de bestelling veelal tijdens uw volgende bijeenkomst in ontvangst kunt nemen.

Belangstelling gevraagd.
Voor het aanvullen van het assortiment in de clubwinkel zouden wij graag willen weten hoe de belangstelling is voor enkele zaken. Zoals daar zijn:
- Big Benny; zie A.N. 1983 nr.1 (met of zonder onderdelenpakket).
- VIA-kaart; voor de schakelaars meerdere VIA's op een print.
- telefoonmodem; om via de telefoon Atom's te koppelen.
- kleurenkaart; kleuren uit de Atom met evt. geluid en 60/50 Hz.- 80 koloms VDU

De tijd tussen deze en de vorige Acorn Nieuws is een beetje te kort geweest om een goed overzicht te krijgen van de "belangstelling". Vandaar deze herhaalde oproep. Wel is al duidelijk dat de interesse voor BB groot is. Deze klok-kaart kunt u dan waarschijnlijk binnenkort in de clubwinkel tegenkomen.
Binnen de Federatie wordt heel wat geknutseld en gebouwd aan en met de Atom. U moet dit dan ook zien als een poging om de producten van de "producenten" wat meer bekendheid te geven, zodat de andere leden daar ook gebruik van kunnen maken.
Wanneer u denkt iets moois ontwikkeld te hebben of mee bezig bent of van plan bent te gaan ontwikkelen, maakt u dat dan even kenbaar dmv een briefkaartje naar Gerhard Visser?
Wanneer u belangstelling hebt voor een of ander project, meldt u dat dan op dezelfde manier?

	R E G I O	voorzitter	secretaris	penn.meester/leden
a	arnhem			J. Uges W. de Withstr. 78 6712 HE Ede 08380-12756 P.2208888
z	zeeland	C. van Driel Walcherseweg 220 4334 NB Middelburg 01180-37246	H. Heringa Bosseweide 32 4431 AE 's Gravenpolder 01103-2437	H. Heringa Bosseweide 32 01103-2437 P.3009914
b	brabant o.	M.L. Verschuren Hosingenhof 30 5625 NL Eindhoven 040-416092	Th. van Kempen het Puyven 71 5672 RB Nuenen 040-836210	F. Ehrlich Roostenlaan 266 5644 BS Eindhoven 040-114183 B.528392794
li	limburg	H. Princen Europalaan W.24 6075 AW Herkenbosch 04752-2534	E. Sanders Korvererf 1 6043 SL Roermond 04750-26051	W. Ernst Cypresstraat 94 6101 JX Echt 04754-5653 P.5457752
be	belgie	J. Lernout Parklaan 4 2540 Hove 03-4553473	J. Myngheer Fr Verdonckstr39-B 1140 Brussel 02-2160404	R. Leyssens Oude Baan 127 3550 Heusden P.954723630149 ACC

	bandjesarchief	datasheets	drukwerkarchief
H.	H. Blijleven	G. Akkermans	F. Monsanto
No	Noordzeestraat 47	Wikke 1	Biesbosch 153
17	1784 BL den Helder	1273 BR Huizen	8032 VD Zwolle
02	02230-16734	02152-60294	

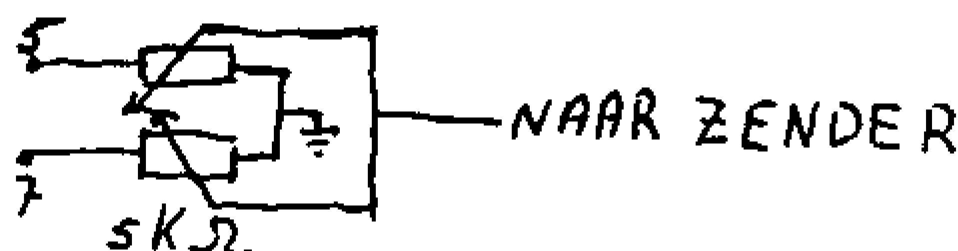
R E G I O	voorzitter	secretaris	penn.meester/leden
noord	F. de Vries Scheldestraat 61 9406 FG Assen 05920-50880	K. Ottes Wilkemaheerd 67 9736 BN Groningen 050-417042	J. de Goede Hunze 26 9406 EE Assen 05920-50675 P.4411898
ov./geld.	J. Ballijns Antaresstraat 12 8303 AJ Emmeloord 05270-14410	G.J. Noorland Pr. Ireneweg 4 7433 DE Schalkhaar 05700-25294	G.J. Noorland Pr. Ireneweg 4 7433 DE Schalkhaar 05700-25294 B.146208234 ACC
twente	M. van Alphen Korianderhof 40 7641 XK Wierden 05496-2264	J. Laarman Potgieterstr. 13A 7422 XP Nijverdal 05486-17178	W. Verhoeven Witbreuksw.385-306 7522 ZA Enschede 053-893674 P.4020195
n. holland	M. de Haan v Hallstraat 537 1051 AC Amsterdam 020-827358	N. Stad Plataanweg 47 1544 PB Zaandijk 075-280808	N. Stad Plataanweg 47 1544 PB Zaandijk 075-280808 B.949672688 ACC
den haag	A. Slootweg Jaarsveldstraat 36 2546 CR den Haag 070-664422	H. vd Heyden Pr. Annalaan 574 2263 XZ L'schendam 070-276726	R. Tiel Vredeoord 96 2544 TZ den Haag 070-294170 P.5467960 ACC
delft	A. Bliek Meerkoetlaan 93 2623 NG Delft 015-611578	M. Schroder Vlamingstraat 42 2611 KX Delft 015-143772	P. Kramer Oudraedweg 36 2612 SN Delft 015-141368 P.5467960 ACC
rotterdam	M. Oosterom Hooidrift 114A 3023 KV Rotterdam 010-255821	J. v Nieuwenhuyzen Brijhoekstraat 32 3223 BE H'voetsl. 01883-15129	R. de Haan Brasem 125 2986 HA Ridderkerk 01804-25160 B.442401094 ACC
centrum	F. Appelman Kastanjelaan 2 4131 AV Vianen 03473-76881	A. Veerman Burg.Reigerstr. 85 3581 KP Utrecht 030-515301	P. van Mourik Ruiterstede 60 3431 XN Nieuwegein 03402-48781 B.300024622 ACC

Het bandjesarchief is uitgebreid met een nieuw bandje.

BANDJE 14:
===========

KANT A
INDEX
MEMOSCOOP maak van de atom een 8 kanaals digitale scoop.
TOEL. SPECTRUM
SPECTRUM spectrum analyser.
LINEAN door rekenen van elektronische schema's zoals filters.
Zie voor de handleiding het volgende AN.
ELEC.1 & ELEC.2 zie bandje 7
ELECTRON schema teken programma.
DIV.PROGRAMMA'S o.a. toongenerator & frekwentiemeter.
WEERSTANDS BER.

KANT B
INDEX
WEERSTAND BER.
CONTEST logboek & QTH-loc.
*RTTY telex zend/ontv. programma
laden op #4000
start link#4000
dan *RTTY
zenden/ontvangen poort B van de VIA
POORT B:
bit 0 ontvangst
bit 1 SQUELCH
bit 2,3,4 vrij
bit 6 T/R switch
bit 5 CALL in MORSE
bit 7 AFSK-output (850 Hz SHIFT)



SHIFT-C Call geven in morse
SHIFT-N Voor eigen toep. bijv. verwijzing naar ander geheugenblok.

RTTY 64/24 zend/ontv. met soft-VDU, link#2CD8;OLD;RUN
FAX 2 ontv. FAX tekeningen van amateurs input is cas.ing. zonder interface.

HELL ontv. amateur HELL stations
PRINTER telex als printer
TELEX idem
PROGRAMMER snel programmeren voor ATOM epromprogrammer
HARMONISCHE laat golfvorm zien door vervorming van harmonische

Dit is dan de inhoud van bandje 14.

Een groep van vier mensen doet met ingang van deze uitgave de redactie van ACORN NIEUWS.
Hierbij zijn in deze groep de taken intern verdeeld. Alle copij dient U aan het redactie adres te sturen. Wij verzoeken de regio-redacteuren twee exemplaren aan de redactie te zenden. Dit bespaart een hoop copieerwerk en kosten.
De regiobladen die door de redactie worden ontvangen zijn helaas niet altijd van die druktechnische kwaliteit dat er zonder problemen vanuit gecopieerd kan worden. Veelal is het orgineel onontbeerlijk.
Het is ons opgevallen dat vele artikelen met een tekst-verwerker zijn gemaakt. We nemen aan dat deze file's door de auteur wel gesaved zijn en als U dit in het verleden niet deed doe het dan nu wel. Want voor publicatie in het landelijke blad zal de redaktie in de toekomst een beroep op de auteurs gaan doen deze bandjes op te sturen. Bandjs die op verzoek van de redaktie worden opgestuurd worden geretourneerd met de gemaakte porto-kosten. Wij hopen op deze manier de leesbaarheid 100% te krijgen.
Verder zullen de gepubliceerde programma's naar het landelijk bandjesarchief worden doorgestuurd zodat de archivaris actueel met zijn archief is.

Wij willen ook met nadruk de auteurs verzoeken zich aan de bladspiegel van 25cm hoog en 17cm breed te willen houden. Daar dit gegarandeerd problemen bij de drukker geeft en verkleinen niet altijd mogelijk is, worden deze artikelen geweigerd.

Wanneer er mensen zijn die de RESET routine in de monitor-rom gewijzigd hebben, klim dan eens in de pen, want dan kunnen we daar een samenvattend artikel over schrijven.

Natuurlijk wil de huidige ploeg, TON OTTEN ook dank zeggen voor zijn eenzame arbeid en voor het prettige kontakt tijdens de "overgave en overname". Hopelijk weet je de nu ontstane leegte in de tijd nuttig te besteden met je bloedeigen ATOMIENTJE.

Wij rekenen op een enthousiaste medewerking van alle clubleden om een interessante en leerzame jaargang 1984 te maken

Frans van Hoesel.
Hans Marks.
Henk Reinders.
Walter-Jan Spaans.

De sluitings datum voor het volgende nummer is 2 APRIL 1984.

Het is een spel dat ik kende als een spel voor de PET. Omdat ik de listing van dat programma niet had kon ik geen vertaling maken. De hier gegeven ATOM versie is dan ook volledig opnieuw geschreven.
Je speelt het spel als volgt:
Je staat aan de rand van een dicht oerwoud en je bent opweg naar je hut aan de andere kant van het bos. Er is echter een moeilijkheid. Er zijn namelijk neushoorns in het bos. Neushoorns zijn tamelijk bijziende en niet zo heel slim maar ook erg argwanend. Als ze de lucht van je krijgen komen ze naar je toe met de zelfde snelheid als waarmee jij beweegt. Dus als jij een pas doet doet de neushoorn dat ook. Pas op want als hij je kan zien,dat is als hij heel dichtbij is,dan wordt hij kwaad en zal je onder de voet lopen. Mocht je toch de hut bereiken dan mag je het nog eens proberen met een neushoorn meer in het bos. Het programma houdt een lijst bij van de beste 10 scores en deelt bonussterren uit.

```
  0REM RHINOCEROS  J.R.MARKS (050-711751)
 10H=0;T=128;L=#9500;M=#9000;T?1=255;$#2840="";A=#FE94
 20P=#2800;C=P;DIMXX100,YY100;P.$21;+LDA#B002;LDY#81;:XX0LDXT
 30:XX1DEX;BNEXX1;EOR@4;STA#B002;DEY;BNEXX0;LDA@255;STA#81;RTS;+
 40P=#2880;B=P;+JSR#FE71;CPY@#FF;BEQB;STYT;RTS;←;P.$6;CLEAR0
 50P.$30'''''''"        *** RHINOCEROS ***"''''"          "
 60P."J.R.MARKS"''"                  (C)1982";LI.A;GOS.c;F.N=1T.1000
 70CLEAR0;P.$30;I.H;I.H%3=0;G.e
 80m@=0;I.H;P."YOU HAVE MADE IT "H" TIME";I.H-1;P."S"
 90I.H;P.".";
100P."THIS TIME "N" RHINO";I.N-1;P."'S"
110P.'"WILL BE IN THE FOREST."';LI.A;CLEAR1;C.1
120F.I=1T.N;XXI=A.R.%64;YYI=A.R.%52+12
130I.XXI)29;I.XXI(34;I.YYI)56;I.YYI(60;I=I-1
140N.;F.I=#8000T.#8200S.4;!I=R.&#AAAAAAAA;N.
150F.I=#8200T.#8280S.4;!I=R.&#AA0A0AAA+#F0F000;N.
160F.I=#8280T.#8300S.4;!I=R.&#0A0A0A0A+#F0F0F0F0;N.
170F.I=#8300T.#8380S.4;!I=R.&#000A0A00+#FFF0F0FF;N.
180F.I=#8380T.#8400S.4;!I=#FFFFFFFF;N.
190F.I=0T.7*16S.16;I?#8027=255;I?#8028=255;N.
200MOVE30,57;DRAW30,59;DRAW33,59;DRAW33,57;DRAW32,57;X=32;Y=2
210hPLOT13,X,Y;R=0;W=Y;V=X;LI.B
220I.?T=57;Y=Y+1;X=X-1;G.b
230I.?T=41;Y=Y+1;X=X+1;G.b
240I.?T=45;X=X+1;Y=Y-1;G.b
250I.?T=34;X=X-1;Y=Y-1;G.b
260I.?T=53;Y=Y+1;G.b
270I.?T=46;Y=Y-1;G.b
280I.?T=43;X=X+1;G.b
290I.?T=39;X=X-1;G.b
300C.2;PLOT13,X,Y;C.1;G.h
310bI.X(0;X=0
320I.X)63;X=63
330I.Y)63;Y=63
340I.Y(0;Y=0
```

```
350E=X;F=Y;GOS.a;X=E;Y=F;?T=17
360I.X=32;I.Y=58;F.I=0T.H;?T=?T+3;LI.C;WAIT;N.;H=H+1;LI.A;N.N
370C.1;PLOT13,X,Y;F.J=1T.N;I.A.(Y-YYJ))100.A.(X-XXJ))10;N.J;G.h
380?T=J*10+100;T?1=10;LI.C;PLOT13,(XXJ),(YYJ);YY0=SGN(Y-YYJ)
390XX0=SGN(X-XXJ);E=XXJ+XX0;F=YYJ+YY0;V=XXJ;W=YYJ
400C.2;PLOT13,V,W;GOS.a;I.G;G.d
410hC.1;PLOT13,E,F;XXJ=E;YYJ=F;I.A.(E-X)(2A.A.(F-Y)(2;G.g
420N.J;G.h
430dI.YY0(1;YY0=YY0+1;G.f
440I.XX0()0;XX0=0;G.f
450I.YY0=1;YY0=0;G.f
460I.XX0=0;XX0=2*(A.R.%2)-1
470fE=XXJ+XX0;F=YYJ+YY0;V=XXJ;W=YYJ;C.2;PLOT13,V,W;GOS.a;G.l
480aZ=E/4+(63-F)*16+#8000;Q=3-E%4;C.2;PLOT13,V,W;G=0
490I.?Z&%(2*4>Q+.5)=0;E=V;F=W;G=1
500R.
510gC.2;PLOT13,X,Y;C.1;PLOT9,1,1;PLOT9,-2,0;PLOT9,0,-2
520PLOT9,2,0;?T=50;LI.C;C.2;PLOT9,0,2;PLOT9,-2,0;PLOT9,0,-2
530PLOT9,2,0;?T=77;LI.C;C.1;PLOT9,1,3;PLOT9,-4,0;PLOT9,0,-4
540PLOT9,4,0;?T=37;LI.C;C.2;PLOT9,0,4;PLOT9,-4,0;PLOT9,0,-4
550PLOT9,4,0;?T=16;LI.C;C.1;LI.A;GOS.i;N=0;H=0;N.N
560cP.$12""RHINOCEROS"""''"IS A GAME IN WHICH YOU HAVE TO"'
570P."WALK THROUGH A FOREST TO GET"'"HOME."';?#E1=0
580P."UNFORTUNATELY THERE ARE RHINO'S IN THIS FOREST."'
590P."WHEN A RHINO SMELLS YOU HE'LL"'"TRY TO FIND YOU."'
600P."AND AS SOON AS HE SEES YOU,"'"WHICH IS WHEN HE'S VERY"
610P." CLOSE,"'"HE'LL GET ANGRY AND ATTACK YOU."';LI.A;P.$12
620?#E1=0;P."THESE KEYS WILL MOVE YOU THROUGHTHE FOREST:"'''
630P."              ↑Y← ↑U← ↑I←"''"              ↑G←          ↑K←"''
640P."              ↑B← ↑N← ↑M←"''';F.I=0T.10;$(M+I*40)="";L?I=0
650N.;$#9600="";P."GOOD LUCK!"';LI.A;R.
660iP.$12"RHINOCEROS"'';@=0
670P."YOU HAVE MADE IT "H" TIME";I.H-1;P."S"
680P."."'"ENTER YOUR NAME."'';IN.$(M+400);L?10=H
690I.L.(M+400))15;?(M+415)=13
700$(M+400+L.(M+400))=$#2840
710jF.I=0T.9;I.I?L(I?(L+1);G.k
720N.;P.$12"RHINOCEROS"'';@=4;?#E1=0
730F.I=0T.9;P.L?I" "$(M+I*40)';N.;P.';LI.A;$#2840=""
740CLEAR0;P.$30"NEXT PLAYER.";LI.A;R.
750k?T=I?L;I?L=I?(L+1);I?(L+1)=?T;$#9600=$(M+I*40)
760$(M+I*40)=$(M+40+I*40);$(M+40+I*40)=$#9600;I=9;N.;G.j
770eF.J=1T.H/3;F.I=240T.21S.-2;?T=I;T?1=5;LI.C;N.
780F.I=210T.240S.2;?T=I;T?1=5;LI.C;N.;N.
790F.J=1T.30;?T=R.;T?1=10;LI.C;N.;?(#283F+H/3)=#2A
800?(#2840+H/3)=13;G.m
```

(Het programma draait op een standaard ATOM en gebruikt geen

P-Charme Interpretor

Anderhalf jaar nadat de eerste bytes geprogrammeerd werden is het dan eindelijk zover: een nieuwe EPROM voor de ATOM. Nu zou u kunnen denken "alweer een utility?" Ja, maar deze is toch wel zeer bijzonder:

1) Een opvallend aantal (zinnige) instructies:

AUTO	BEEP	BSAVE
CASE...OF, CEND	CAT	STOP, CONT
COPY	COS1, COS2	DATA, READ, RESTORE
DEL	XIF...ELSE	EVEN, ODD
TRUE, FALSE	FUNCTION, FEND	HEX
HTAB, VTAB	ICOPY	KEY, INKEY
INSTR	NOT	ON ERROR
ON...GOSUB	ON...GOTO	PAUSE
PROC, PEND	POP	PROGRAM
RENUM	VAR	WHILE, WEND
ZERO		

2) Multi-dimensionale (floating-point) array's in de vorm van bv.

```
DIM KK(10,2,2)
SS(1,1,12,3)=24
PRINT DD(1,2,1)'
%A=%FF(8,3)/2.5
```

Hierbij worden alle indèces gecontroleerd met de opgegeven waarden in het bijbehorende DIM statement. Indien dat niet klopt volgt een error 134: 'Array subscript out of range'.

3) Een ingebouwde error-aanwijs-routine. Deze geeft met behulp van een sterretje boven de regel de plaats aan waar de error gesignaleerd werd.

4) Modelijkheid om met het PROGRAM statement een naam aan een programma te geven, waardoor die naam automatisch herkend wordt als statement. De naam van een programma als statement zorgt ervoor dat het betreffende programma wordt opgezocht in het geheugen en correct(!) wordt opgestart (TOP wordt dus eerst opgezocht, labels worden geinitialiseerd, FOR-NEXT-loops worden gereset, etc).

5) een ingebouwde editor die het wijzigen van programma's wel heel erg gemakkelijk maakt.

6) Een haast onbeperkt aantal statements kan door de programmeur worden bijgemaakt en in RAM worden geplaatst. De P-Charme interpreter (spreek uit: pii-tcharmm) zal die statements herkennen en uitvoeren, en er is dus geen extra interpreter of schakelkaart nodig om de ATOM extra statements te leren.

De in P-Charme opgenomen statements zijn onder andere geselecteerd op een grote mate van systeem onafhankelijkheid. Een uitzondering hierop vormt echter het COS statement(1200 Baud en de stille passages tussen de afzonderlijke blokken worden verwijderd). Dit statement is niet interessant voor disk-systemen, maar is toch opgenomen in P-Charme opdat deze EPROM een waardig opvolger van de 'JOSBOX' zou kunnen worden.

Bij de P-Charme interpretor behoord een zeer fraaie (gedrukte !) handleiding waarin alle statements uitvoerig en duidelijk worden beschreven. Dit drukwerkje is samen met de software waarschijnlijk ook in uw regio leverbaar tgen een minimale vergoeding van ongeveer f 4.75.

Alle in deze EPROM opgenomen statements zijn door de auteur zelf geprogrammeerd op ongeveer 600 velletjes kladpapier en tot nu toe zijn, ondanks uitvoerige test-procedures, nog geen fouten gevonden.

Voor de kenners alvast een voorproefje van wat met P-Charme mogelijk is:

```
 10 PROGRAM ACKERMANN
 20
 30 FUNCTION ACK(M,N)
 40  IF M=0 ACK=N+1;GOTO a
 50  IF N=0 ACK=ACK(M-1,1);GOTO a
 60  ACK=ACK(M-1,ACK(M,N-1))
 70aFEND
 80
 90 PRINT ACK(2,3)'
100 END
```

De procedures en functies hebben o.a. de volgende 'eigenschappen':

volledige recursie (tot het geheugen vol is)
lokale variabelen
value parameters
var parameters
meer-letterige namen

FRANS VAN HOESEL

De in HET BRONSGROEN EIKELTJE 2 beschreven karaktergenerator maakte het mogelijk om ook kleine letters weer te geven. Maar daardoor viel het cursor effect (inventeren) weg. Als de cursor nu over hoofdletters schoof werden het kleine letters en omgekeerd. Een ander nadeel was dat alle grafics modes van 0 tot 4 vervielen. De door ons ontworpen karaktergenerator heeft deze nadelen niet meer en het is nu zelfs mogelijk om de oude karakterset met kleine letters, uit te breiden met nog 256 eigen te ontwerpen karakters.

Om de cursor normaal te laten werken moet ?#E1 niet #80 maar #40 zijn. Bij gebruik van grafic mode 0 moet de karaktergenerator omgeschakeld worden. Dit kan bijvoorbeeld worden gedaan door een output bit van de VIA of PIA te gebruiken, bijv. in de tekening is bit 3 van#B002 gebruikt. Bij het prototype werd via een wire wrap de print bevestigd in de IC voet van de 6847 VDU controler. Daardoor kwam deze print tussen de behuizing van de Acorn en de IC's op de computerprint te zitten. De 2532 eprom van de karaktergenerator wordt continue gebruikt en wordt daardoor heter dan normaal. Daarbij komt nog de warmte die de IC's van de computer afgeven en door de kleine ruimte kan deze niet weg. Het gevolg hiervan was storingen. Deze storingen hebben we verholpen door een koelplaat over de gehele karkatergenerator aan te brengen.

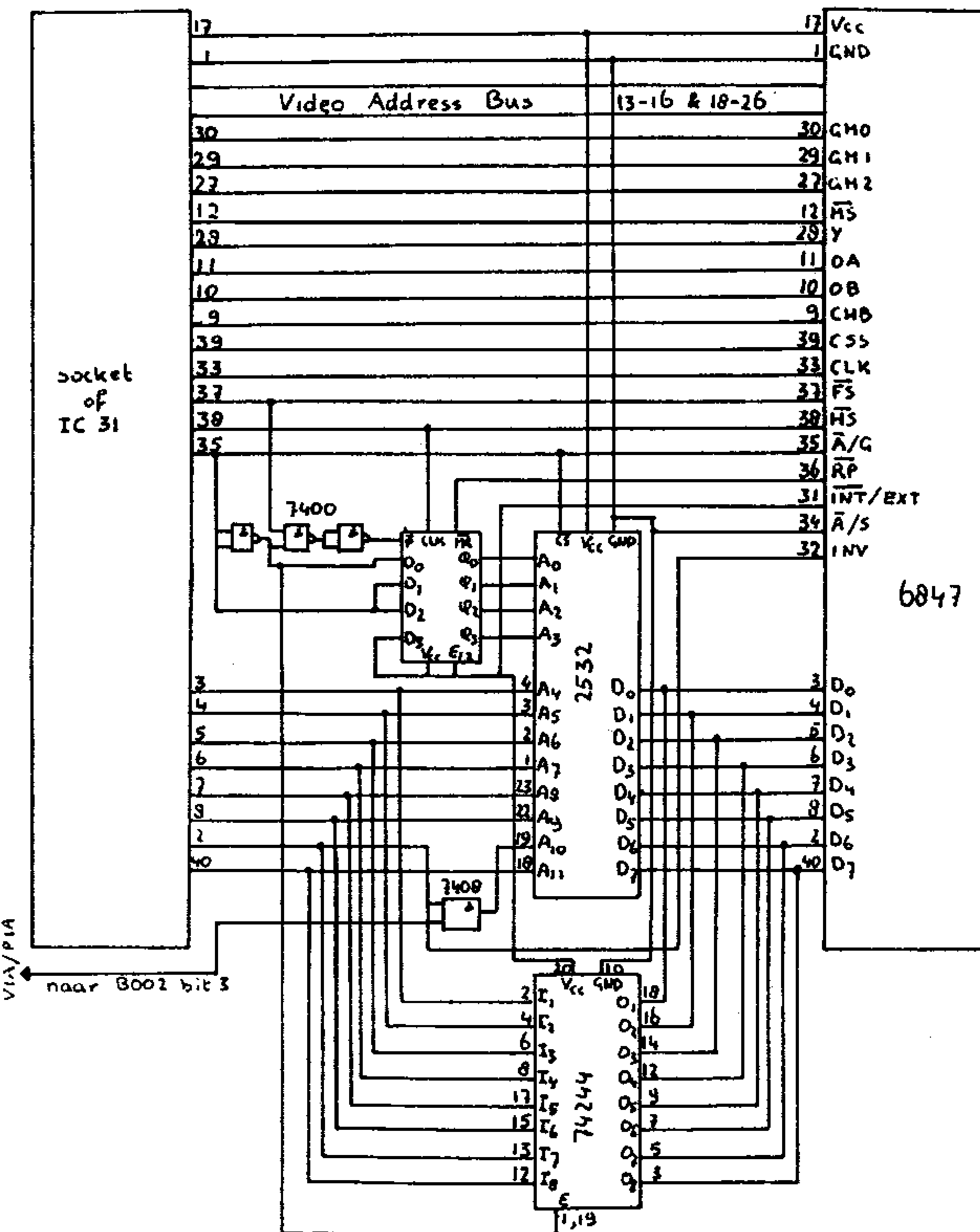
Het programeren van een karakter gaat nu als volgt.
bijvoor beeld de hoofd letter A:

```
geheugenadres XXX0 = #00 )
geheugenadres XXX1 = #00 } ruimte boven de letter
geheugenadres XXX2 = #00 )
geheugenadres XXX3 = #1C )
geheugenadres XXX4 = #22 |
geheugenadres XXX5 = #02 |
geheugenadres XXX6 = #1A |
geheugenadres XXX7 = #2A } de letter zelf
geheugenadres XXX8 = #2A |
geheugenadres XXX9 = #1E )
geheugenadres XXXA = #00
geheugenadres XXXB = #00
geheugenadres XXXC = #F8 )
geheugenadres XXXD = #07 |
geheugenadres XXXE = #F8 } mapping control codes
geheugenadres XXXF = #07 )
```

De mapping control codes moeten onder iedere letter geplaats worden.

Het bandje met de nieuwe karakters is verkrijgbaar via het bandjes-arcief.

Schema voor de nieuwe karaktergenerator.




```
?#E1=40
GMO; ?#B002=?#B002:8
```

Josbox verbeteringen

Hoewel de Josbox een zeer handige en nuttige toolkit is, zijn er een paar minpuntjes die in de praktijk storend werken. De hier voorgestelde wijziging van een paar bytes levert het volgende op:

- Ook bij SCOS korte pauzes tussen de blokken.
  Wijziging: A51B:E7 A5

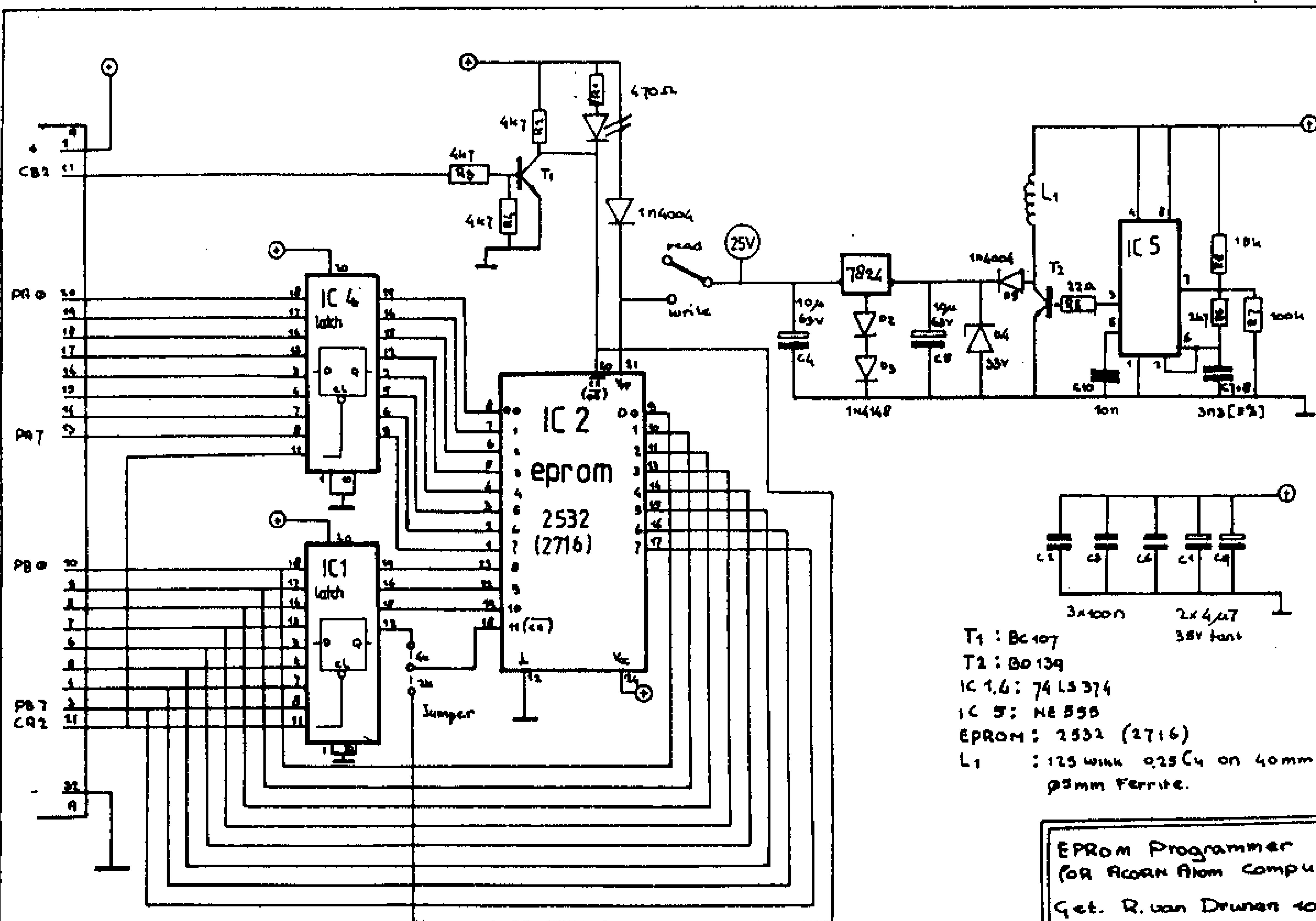
- Altijd grijze cursor in tekstmode bij FCOS en SCOS.
  Momenteel wordt de cursor alleen grijs als de OSWRCH-vector op #FE52 staat. (Er wordt gecontroleerd of het high byte ervan #FE is.)
  Het zou beter zijn de cursor altijd grijs te maken als het scherm in tekstmode staat (dus niet in GRMOD o.i.d.).
  Wijziging: A575:00 B0
  A578:00

- Iets kortere pauze tussen het begin van een blok en de eigenlijke data bij FCOS en SCOS. De pauze wordt nu net zo kort als bij ALLCOS van A.Marchal (zie AcornNieuws 1.8).
  Wijziging: A667:06

- Correct werkend FIND statement.
  Bij het FIND statement worden de regels die de opgegeven string bevatten met negatief regelnummer afgedrukt als vlak tevoren een negatief getal in de Basic accu is verwerkt.
  Wijziging: A2AD:89

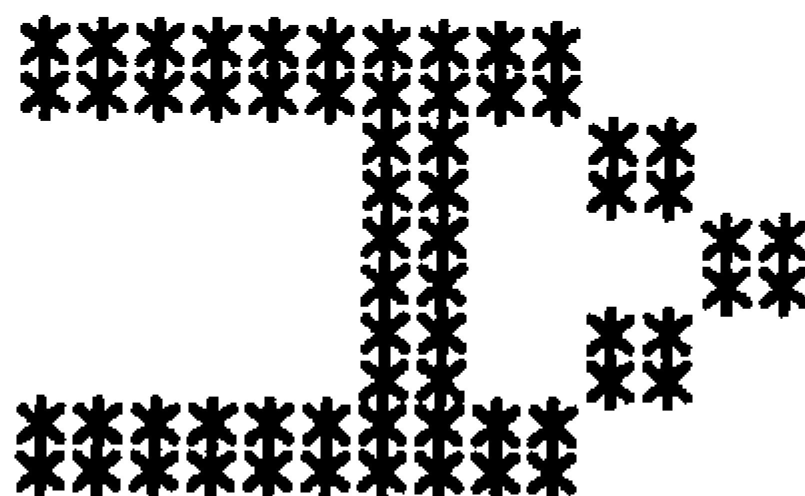
- Verbeterd XDUMP commando.
  Bij XDUMP wordt de Ascii code #7F (=Delete) ook afgedrukt, waardoor de Ascii dump er niet meer netjes uitziet.
  Wijziging: A6D5:FF AB C9 7F B0
  ABFF:20 2A A7 B1 92 60

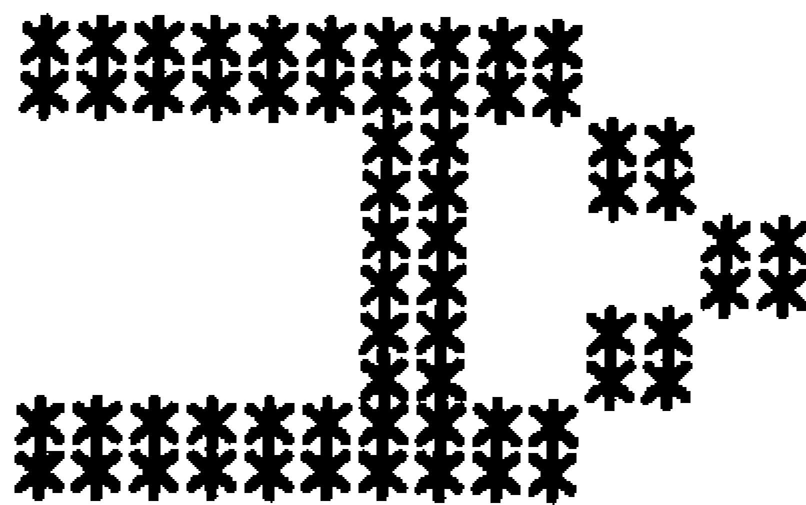
Indien U bovenstaande wijzigingen (laat) aanbreng(t)(en) in Uw Josbox blijft U volledig compatible met de 'normale' Josbox.
Als iemand andere foutjes heeft ontdekt en/of wijzigingen heeft bedacht, kan hij/zij dat melden bij de redactie.


IC 4
latch
IC1
latch
IC 2
eprom
2532
(2716)
IC 5
7824
PA 0
PA 7
PB 0
PB 7
CB2
CA2
read
write
25V
470Ω
4k7
1N4004
1N4148
T1
T2
L1
Jumper
3x100n
2x4µ7
35V tant
T1 : BC107
T2 : BD139
IC 1,4 : 74 LS 374
IC 5 : NE 555
EPROM : 2532 (2716)
L1 : 125 wikk 0,25 Cu on 40mm
ø5mm Ferrite.
EPRom Programmer
for Acorn Atom Computer
Get. R. van Drunen 10/02/8

SPANDOEK

Dit programma is geschreven om in combinatie met een Seikosha GP-250X te werken.
Het programma vraagt een naam of tekst van max. 32 tekens, en zet deze vervolgens in groot formaat op papier.Om dit programma te laten werken dient de josbox geplaatst te zijn.
De letter A zal dan zo uitgeprint worden:





```
   5 REM spandoek
   6 REM p.lauwen
   7 REM amarildijk 117 roosendaal
   8 REM Tel.01650-49310
  10 R=#FFF4
  20 DIMA32
  30 P.$12
  40 IN."NAAM"$A
  50 GR.;?#E1=0
  59 P.$A
  60 B=LEN(A) ;?#88=B
 100 P.$21;P.$3
 110 DIMLL20
 120 F.X=0 TO20;LLX=-1;N.
 130 F.X=0 TO 1
 140 P=#3000
 150[LDA@ 2JSRR ;LDA@7;JSRR;LDA@13;JSRR
 160 LDA@2;JSRR;LDA@27;JSRR;LDA@66;JSRR
 165 LDA@27;JSRR;LDA@76;JSRR;LDA@#81;JSRR
 170 LDA@#60;STA#80;LDA@#80;STA#81;LDA@2;STA#87
 190:LL0LDY@0;:LL1 LDX@8;:LL2 LDA(#80),Y;STA#2800,X
 200 CLC;LDA#80;ADC@#20;STA#80
 205 CPX@4;BEQLL13;DEX;BNELL2
 206 JMPLL14
 207:LL13 INC#81;DEX;JMPLL2
 208:LL14DEC#81
 210 LDA@8;STA#84;:LL4 LDX@8
 220:LL3  ;ROL#2800,X;ROR#83;DEX;BNELL3
 250:LL6 LDA@8;STA#85
 255 LDA#83;STA#86
 260:LL10 CLC
 270 ASL#86;BCCLL11
 275 LDA@#2A;JSRR;JSRR;JMPLL12
 280:LL11 LDA@#20;JSRR;JSRR
 290:LL12 DEC#85;BNELL10
 295 LDA@13;JSRR;JSRLL15;LDA@13;JSRR;JSRLL15
 300  NOP  ; NOP  ;DEC#87;BNELL6;LDA@2;STA#87
 303 DEC#84;BNELL4
 305 INY;CPY #88;BNELL1;RTS
 308:LL15LDA@27;JSRR;LDA@16;JSRR;LDA@0;JSRR;LDA@126;JSRR;RTS;]
 310 N.
 315 GR.;?#E1=0;P.$A
 320 LINK#3000
 330 P.$7$15$3
 335 P.$6
 340 P.''''
 345 P."     klaar"
 350 E.
```

```
 10 REM:A.PETERS WEHL.
 20 REM:TEKENT BEKENDE TV
 30 REM:FIGUUR OP MONITOR.
 40 REM:LEUK VOOR KLEINE
 50 REM:KINDEREN (OOK GROTE).
 60 REM:SCHAKEL JOSBOX IN,
 70 REM:ANDERS REGEL 400 EN
 80 REM:410 WEGLATEN.
100 CLEAR4;MOVE128,144;DRAW104,72
110 DRAW104,48;DRAW128,24;DRAW152,48
120 DRAW152,72;DRAW176,120;DRAW152,144
130 DRAW152,156;DRAW164,144;DRAW152,144
140 MOVE164,132;DRAW176,144
150 DRAW152,168;DRAW128,144;DRAW152,72
160 DRAW104,72;DRAW80,120;DRAW104,144
170 DRAW92,144;DRAW104,156;DRAW104,144
180 MOVE92,132;DRAW80,144
190 DRAW104,168;DRAW128,144
200 MOVE120,120;DRAW92,120
210 DRAW104,104;DRAW116,120
220 MOVE136,120;DRAW164,120
230 DRAW152,104;DRAW140,120
240 MOVE152,72;DRAW128,60;DRAW104,72
250 MOVE120,56;DRAW64,56
260 MOVE136,56;DRAW192,56;MOVE144,64;DRAW184,72
270 MOVE112,64;DRAW72,72;MOVE104,48;DRAW152,48
280 MOVE103,117;DRAW104,117;MOVE151,117;DRAW152,117
290 MOVE102,116;DRAW105,116;MOVE150,116;DRAW153,116
300 MOVE101,115;DRAW106,115;MOVE149,115;DRAW154,115
310 MOVE100,114;DRAW107,114;MOVE148,114;DRAW155,114
320 MOVE100,113;DRAW107,113;MOVE148,113;DRAW155,113
330 MOVE101,112;DRAW106,112;MOVE149,112;DRAW154,112
340 MOVE102,111;DRAW105,111;MOVE150,111;DRAW153,111
350 MOVE103,110;DRAW104,110;MOVE151,110;DRAW152,110
400 PLAY E4,F4,R2,G4,G#4,R2,E4,F4,R4,G4,G#4,R4,C'#4,C'4,R4
410 PLAY F4,G#4,C'4,B'1,A'#4,G#4,F4,D#4,E4,F4,R1,F'4
500 FOR X = #8000 TO #97FF;?X=255;NEXT
510 REM:VERANDER 255 IN REGEL
520 REM:500 VOOR ANDER EIND
600 END
```

Voor hen die er niet uitkomen.

Spelletjes op de ACORN is best wel eens leuk.
Nu ben ik zelf niet zo'n spelletjesmaniak, maar een enkele keer houd ik mij ook hiermee bezig.
Een erg leuk spel vind ik "3D-Maze" uit het ATOM MAGIC BOOK.
Toch heeft het spel m.i. een nadeel.
Zitten we namelijk in een groot doolhof (b.v. 18 x 18) en zijn we na een half uur sjouwen de kluts kwijt, dan is er de toets H.
Het programma geeft ons dan keurig de coordinaten en zelfs de richting waarin we vertwijfeld staren.
Helaas, zonder plattegrond doe ik daar niets mee!
Daarom heb ik het programma een klein beetje uitgebreid.
In de eerste plaats willen wij het plattegrond bewaren.
Dit doen we met een subroutine op b.v. regel 5000.
De subroutine roepen we aan in regel 702 (....dus toevoegen).

```
702 GOSUB 5000
```

De subroutine zelf is erg eenvoudig èn behoeft daarom geen uitleg.

```
5000 FOR S=0 TO 512
5010 S?#9600=S?#8000
5020 NEXT
5030 RETURN
```

Om d.m.v. toets H de plattegrond weer te bekijken veranderen we regel 1129 in het programma.

```
1129 GOSUB 6000;GOTO p
```

De subroutine op regel 6000 doet in de eerste plaats het omgekeerde van subr. 5000 (reg. 6010-6030).
Daarna maken we onze netelige positie zichtbaar met MOVE en PLOT.
Dit gebeurt net zolang totdat wij de CTRL-toets hebben ingedrukt.

```
6000 P."DRUK OP CTRL-TOETS"'
6010 FOR S =0 TO 416
6020 S?#8000=S?#9660
6030 NEXT
6040 MOVE(X+1),Y;PLOT 10,(X+1),Y
6050 IF ?#B001<>#FF RETURN
6060 WAIT;WAIT;WAIT
6070 MOVE(X+1),Y;PLOT 10,(X+1),Y
6080 WAIT;WAIT;WAIT
6090 GOTO 6040
```

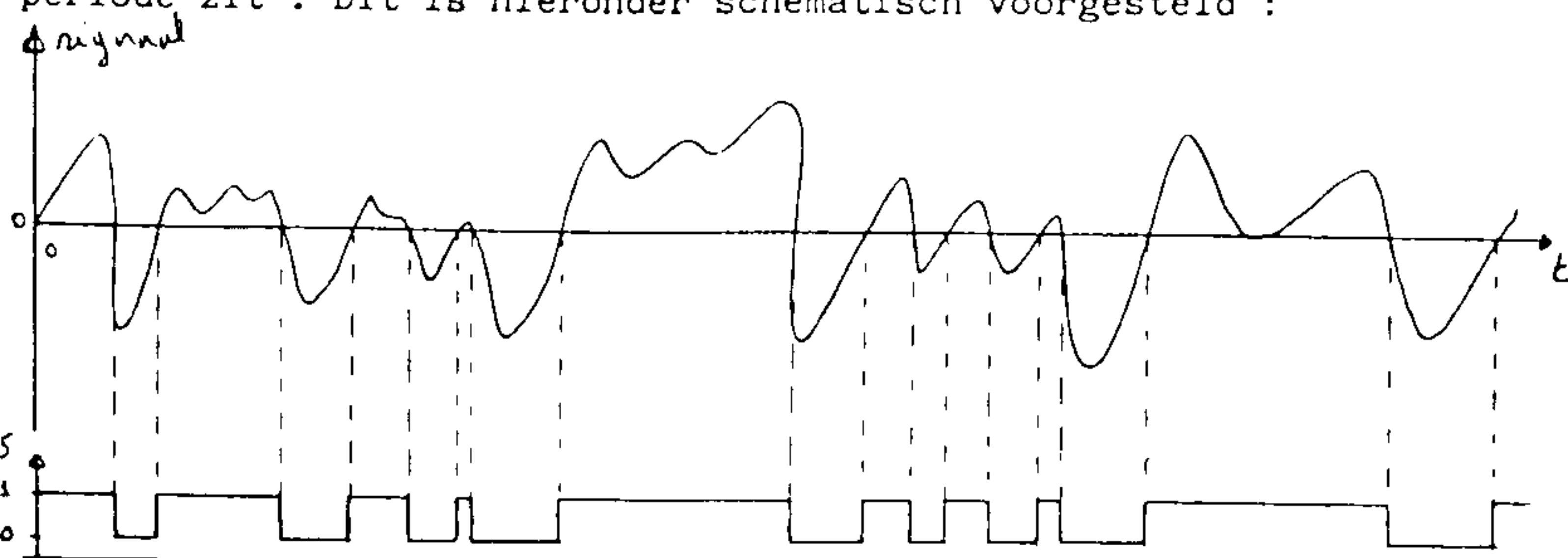
Met deze eenvoudige toevoegingen wordt ook de moeilijkste doolhof begaanbaar.

De 6502 in onze ATOM krijgt een opvolger en wel de 65C02 en de R65C02. Deze beide processors zijn pincomptabel, d.w.z. dat ze dezelfde pinaansluiting hebben als de 6502 en zodoende zo in onze ATOM geprikt kunnen worden. In ELEKTUUR van oktober staat wat de verschillen zijn met de 6502. Daarin staat ook een tabel afgedrukt met de toegevoegde instruktie's. Aan de hand hiervan is een complete nieuwe tabel gemaakt van de R65C02, welke hierbij wordt afgedrukt. Bij nadere bestudering van ELEKTUUR blijkt dat er een belangrijk verschil is tussen de R65C02 en de 65C02. Dit verschil zit hem in 4 intruktie's meer in de R65C02. Dit zijn de intruktie's BBR, BBS, RMB en SMB. Bij nadere informatie bleek mij dat de 65C02 reeds te koop is en dat de R65C02 in november leverbaar wordt. Er zijn nog meer type's zoals de R65C102, R65C112 maar deze zijn niet pincomptabel en moet er dus in de ATOM gesoldeerd worden. De mogelijkheden van deze procesoren zijn weer groter en ingewikkelder.
Ik wil mij verder nu beperken tot het noemen van de nieuwe instrukties en het uiteenzeten van een nieuwe adresseringsmetode (de 14e).
Eerst dan nu oude bekende instruktie's die nu uitgebreid zijn tot bekende adresseringsmetoden.
ØRA (ind). SBC (ind). STA (ind). BIT imm, z,x abs,x
JMP ind,x. ADC (ind). AND (ind). EOR (ind.) CMP (ind).
LDA (ind)
Dan nu nieuwe instruktie's:
DEA verlaag de accumulator met 1.
INA verhoog de accumulator met 1.
PHX zet X-register op stack.
PHY zet Y-register op stack.
PLX haal x-register van stack.
PLY haal Y-register van stack.
BRA spring altijd (relatief)
STZ maak de opgegeven geheugenplaats nul. (abs z z,x abs,x)
TRB voert een BIT instuktie uit en zet resultaat terug op geheugenplaats.
TSB zelfde als TRB maar dan doormiddel van een logische or-functie inplaats van een and-funtie.
RMB zet opgegeven bitje van zero-page adres Ø. bijv. RMB3 #8Ø.
SMB zet opgegeven bitje van zero-page adres 1. bijv. SMB6 #8Ø.
BBR spring relatief indien opgegeven bitje Ø is. (BBR3 #8Ø,#2814)
BBS spring relatief indien opgegeven bitje 1 is. (BBS6 #8Ø,#2819)
Zoals men kan zien is dit een belangrijke uitbreiding voor onze ATOM en loont het zeker het wachten op de uitgebreidere R65C02.
Uiteraard moet voor deze nieuwe processor ook een nieuw assembleer programma worden geschreven. Hierbij is het mij opgevallen dat niet alle nieuwe opcodes een logische waarde konden krijgen ten opzichte van het absolute adres. Zodoende is het oude programma in de MONITOR-rom van #F291 - #F53Ø niet uit te breiden maar moet opnieuw op een andere methode worden geschreven.

Ø	BRK IMP	4Ø	RTI	8Ø	BRA REL	CØ	CPY IMM
1	ORA IND,X	41	EOR IND,X	81	STA IND,X	C1	CMP IND,X
2	*	42	*	82	*	C2	*
3	*	43	*	83	*	C3	*
4	TSB Z	44	*	84	STY Z	C4	CPY Z
5	ORA Z	45	EOR Z	85	STA Z	C5	CMP Z

6	ASL Z	46	LSR Z	86	STX Z	C6	DEC Z
7	RMB0 Z	47	RMB4 Z	87	SMB0 Z	C7	SMB4 Z
8	PHP IMP	48	PHA IMP	88	DEY IMP	C8	INY IMP
9	ORA IMM	49	EOR IMM	89	BIT IMM	C9	CMP IMM
A	ASL A	4A	LSR A	8A	TXA IMP	CA	DEX IMP
B	*	4B	*	8B	*	CB	*
C	TSB ABS	4C	JMP ABS	8C	STY ABS	CC	CPY ABS
D	ORA ABS	4D	EOR ABS	8D	STA ABS	CD	CMP ABS
E	ASL ABS	4E	LSR ABS	8E	STX ABS	CE	DEC ABS
F	BRR0 ZREL	4F	BRR4 ZREL	8F	BBS0 ZREL	CF	BBS4 ZREL
10	BPL REL	50	BVC REL	90	BCC REL	D0	BNE REL
11	ORA IND,Y	51	EOR IND,Y	91	STA IND,Y	D1	CMP IND,Y
12	ORA IND	52	EOR IND	92	STA IND	D2	CMP IND
13	*	53	*	93	*	D3	*
14	TRB Z	54	*	94	STY Z,X	D4	*
15	ORA Z,X	55	EOR Z,X	95	STA Z,X	D5	CMP Z,X
16	ASL Z,X	56	LSR Z,X	96	STX Z,Y	D6	DEC
17	RMB1 Z	57	RMB5 Z	97	SMB1 Z	D7	SMB5 Z
18	CLC IMP	58	CLI IMP	98	TYA IMP	D8	CLD IMP
19	ORA ABS,Y	59	EOR ABS,Y	99	STA ABS,Y	D9	CMP ABS,Y
1A	INA A	5A	PHY IMP	9A	TXS IMP	DA	PHX IMP
1B	*	5B	*	9B	*	DB	*
1C	TRB ABS	5C	*	9C	STZ ABS	DC	*
1D	ORA ABS,X	5D	EOR ABS,X	9D	STA ABS,X	DD	CMP ABS,X
1E	ASL ABS,X	5E	LSR ABS,X	9E	STZ ABS,X	DE	DEC ABS,X
1F	BRR1 ZREL	5F	BRR5 ZREL	9F	BBS1 ZREL	DF	BBS5 ZREL
20	JSR ABS	60	RTS IMP	A0	LDY IMM	E0	CPX IMM
21	AND IND,X	61	ADC IND,X	A1	LDA IND,X	E1	SBC IND,X
22	*	62	*	A2	LDX IMM	E2	*
23	*	63	*	A3	*	E3	*
24	BIT Z	64	STZ Z	A4	LDY Z	E4	CPX Z
25	AND Z	65	ADC Z	A5	LDA Z	E5	SBC Z
26	ROL Z	66	ROR Z	A6	LDX Z	E6	INC Z
27	RMB2 Z	67	RMB6 Z	A7	SNB2 Z	E7	SMB6 Z
28	PLP IMP	68	PLA IMP	A8	TAY IMP	E8	INX IMP
29	AND IMM	69	ADC IMM	A9	LDA IMM	E9	SBC IMM
2A	ROL A	6A	ROR A	AA	TAX IMP	EA	NOP IMP
2B	*	6B	*	AB	*	EB	*
2C	BIT ABS	6C	JMP IND	AC	LDY ABS	EC	CPX ABS
2D	AND ABS	6D	ADC ABS	AD	LDA ABS	ED	SBC ABS
2E	ROL ABS	6E	ROR ABS	AE	LDX ABS	EE	INC ABS
2F	BRR2 ZREL	6F	BRR6 ZREL	AF	BBS2 ZREL	EF	BBS6 ZREL
30	BMI REL	70	BVS REL	B0	BCS REL	F0	BEQ REL
31	AND IND,Y	71	ADC IND,Y	B1	LDA IND,Y	F1	SBC IND,Y
32	AND IND	72	ADC IND	B2	LDA IND	F2	SBC IND
33	*	73	*	B3	*	F3	*
34	BIT Z,X	74	STZ Z,X	B4	LDY Z,X	F4	*
35	AND Z,X	75	ADC Z,X	B5	LDA Z,X	F5	SBC Z,X
36	ROL Z,X	76	ROR Z,X	B6	LDX Z,Y	F6	INC Z,X
37	RMB3 Z	77	RMB7 Z	B7	SMB3 Z	F7	SMB7 Z
38	SEC IMP	78	SEI IMP	B8	CLV IMP	F8	SED IMP
39	AND ABS,Y	79	ADC ABS,Y	B9	LDA ABS,Y	F9	SBC ABS,Y
3A	DEA A	7A	PLY IMP	BA	TSX IMP	FA	PLX IMP
3B	*	7B	*	BB	*	FB	*
3C	BIT ABS,X	7C	JMP IND,X	BC	LDY ABS,X	FC	*
3D	AND ABS,X	7D	ADC ABS,X	BD	LDA ABS,X	FD	SBC ABS,X
3E	ROL ASB,X	7E	ROR ABS,X	BE	LDX ABS,Y	FE	INC ABS,X
3F	BRR3 ZREL	7F	BRR7 ZREL	BF	BBS3 ZREL	FF	BBS7 ZREL

Een speech-synthesizer is zoals de naam reeds zegt een apparaat dat de spraak synthetiseert .Kunstmatige spraak dus . Hoe doen we zoiets ? Om dat te begrijpen moet men eerst inzien dat spraak samengesteld is uit verschillende klanken In elektrische gedaante is dit een bepaald golfpatroon van het signaal . Dit elektrisch signaal , verkregen uit bijv. een mikrofoon stelt exact (op vervorming en dynamisch bereik na natuurlijk) het geluidssignaal voor . Met dit signaal kunnen we immers steeds terug geluid maken , desnoods na een versterking Indien we dit signaal dus ergens in een geheugen opslaan , kunnen we het later terug gebruiken . Zo'n geheugen waar we een analoog signaal kwijt kunnen is bijvoorbeeld een ordinaire plaat of een magneetband . Ook onze Atom echter kan zulk signaal stockeren . Het moet daarvoor natuurlijk wel eerst het signaal ve talen in Ø of 1 of kombinaties daarvan . Een mogelijkheid is om met een ADC het signaal te vertalen in een 8-bits binair getal en dit getal op verschillende tijdstippen in het geheugen te bewaren . Het aantal malen dat men per sekonde het signaal in het geheugen opslaat noemt men de sampling-frekwentie . Nu bestaat er een theorie die zegt dat de sampling frekwentie 2 maal hoger moet zijn dan de hoogste frekwentie die men ongehinderd wil registreren . Dit is het zogenaamde "samplingtheorema van Shannon" . Als we dus spraak tot 300C willen opnemen dan moeten we sampellen met 6000 Hz .Dit betekent 6000 bytes pe sekonde spraak ! Een 16k kaart is in 2,66 sekonden vol ! Dit is dus niet doenbaar . Om de spreektijd te verlengen kunnen we het systeem gebruiken uit het programma TALKBACK dat in in de MAGIC BOOK staat (p.14) . Dit programm werkt als volgt : iedere 125 mikrosekonden (8000 keer per sekonde) wordt de in houd van bit 5 van poort C uit de 8255 gelezen . Dit is de cassette ingang . Deze is Ø of 1 naargelang het casettesignaal in de negatieve-of positieve periode zit . Dit is hieronder schematisch voorgesteld :



Dit is natuurlijk een zeer grove benadering , maar blijkt in de praktijk wel te werken mits de nodige brom en ruis . Het lijkt mij zeer leerrijk om voor zichzelf eens dit programma uit te proberen , het werkt echt . In verband met geheugen gebruik hetvolgende : het programma behoeft 8000 bits/sek ,dus 1000bytes/sek. Dit is 1/6 de van de vorige metode.

De in de handel te verkijgen systemen doen het echter iets anders . Ten eerste slaagt men erin door zorgvuldig bereken en testen de geheugenhoeveelheid die 1 sekonde spraak gebruikt gevoelig te verkleinen . Dit gebeurt door alle "overtollige" informatie uit het signaal te verwijderen , men laat alleen dat over dat absoluut nodig is om de spraak nog te verstaan . Wat overblijft is iets dat nog weinig van een menselijke stem weg heeft , maar toch nog verstaanbaar is. Ten tweede gaat men met een soort van tweede processor werken . Dit is een speciaal IC met speciale spraakfaciliteiten . In de opzet van dit IC zijn er twee tendenzen te bespeuren . Vooreerst zijn er de woordgenerators , anderzijds de foneemgenerators .

Bij de woordgenerator moet de Atom (in dit geval) het woord dat hij ten gehore wil brengen doorsturen naar het spraak IC . Dit IC zal dan , met behulp van de nodige stuurschakelingen gaan kijken of het betreffende woord in zijn vocabularium voorkomt . Is dit het geval dan zal hij het woord uitspreken met behulp van de code die in zijn "woordenboek" staat ,anders zal hij zich in stilzwijgen hullen . Een dergelijk systeem is bijv. dat van Elektuur dat met een IC van TI werkt . Het nadeel van woordgenerators is dat het systeem vrij komplex is omdat het spreek IC nogal wat besturingslogica nodig heeft . De prijs is nogal hoog en bovendien is men beperkt tot de woorden die de fabrikant zijn IC meegeeft . De woordenschat van het IC zit immers in 1 of meerdere ROM's (meestal Engels woorden) die aan het systeem toegevoegd worden . Daar echter het spraak IC toch nog zo'n 1200 à 2000 bits/sek nodig heeft (die staan in zijn ROM's) lijkt het mij dus wel duidelijk dat het hier onmogelijk is om zelf woorden te vormen . Aldus : exit woordgenerator .

Bij de foneemgenerators werkt men anders . Zoals de naam het zegt kan dit IC klanken generen . Zo'n IC is ondermeer de SC-01 (er bestaat ook al een SX-01) . Dit IC van VOTRAX kan 61 klanken genereren . Deze omvatten alle klanken die in het Engels gebruikt worden . Er zijn natuurlijk meer klanken dan dat er letters zijn om dat 1 letter op verschillende manieren uitgesproken kan worden . Aldus zijn er verschillende e's : é,è,ê,ë,.... mijn fantasie schiet mij te kort . Om dat er nog 3 extra ,niet uit te spreken "stop"klanken zijn om de foneemgenerator wijs te maken wanneer hij lang genoeg gezeverd heeft , zijn er 64 mogelijkheden.

Hiervoor volstaan 6 datalijnen . De twee overige worden gebruikt voor instelling van de pitch . Daarnaast zijn er nog drie aansluitingen om het IC te sturen met handshaking of interrupt . Het gehele systeem is vreselijk eenvoudig ; het bestaat uit de SC-01 plus een versterker plus enkele CMOS poortjes om de TTL'niveaus van de computer aan te passen aan de voedingsspanning van de SC-01,die tussen 7 en 12 volt mag liggen . Dit systeem kan direkt vanuit een VIA poort gestuurd worden (bijv.:PBØ tot PB7 plus CB1 en CB2) en kan elke Engelse (of Nederlands met een Engels accent) tekst spreken door de juiste klanken op de juiste plaats te zetten . Omdat 1 klank gemiddeld ca 85 ms duurt hebben we zo'n 12 klanken per sekonde . We gebruiken dus slechts 12 bytes per sekonde spraak . Op een 16k kaart past dus zo'n 20 minuten spraak . Dat is reeds een grote verbetering . Bovendien zijn we niet gebonden aan de woordenschat geleverd door de fabrikant omdat we direkt zelf woorden kunnen samenstellen . De prijs van dit systeem ligt ook aanmerkelijk lager dan het vorige . Het gebruik van dit systeem zal door Jan Lernout en mijzelf onderzocht worden van zodra wij onze bestelling uit Engeland door de administratieve mallemolen van de douane geloodst hebben . We zullen ons best doen om al het een en het ander te tonen in Rendeux , maar voorlopig ligt het initiatief nog bij de douane .

```
 10REM LISSAJOUS
 20REM HOENEN BRUNSSUM
 30GRMOD;REM JOSBOX AAN
 40DIMA1
 50%A=0;X=128;Y=96;%R=0;%S=0
 60IN."WILT U PUNTEN "$A
 70IF?A=#4A;K=9
 80IF?A=#4E;K=2
 90P.$12;?#E1=0;P."LISSAJOUS"
100F.I=1TO400
110PLOT4,X,Y
120%R=100*SIN(2*%A)
130%S=84*COS(3*%A)
140PLOTK,%R,%S
150%A=%A+1
160N.;E.
```

EEN ANDERE NAAM HEB IK ER NIET VOOR KUNNEN BEDENKEN. HET GAAT HIER ECHT OM EEN KLEIN PRINTJE MET GROTE MOGELIJKHEDEN.
TEN EERSTE ZIJN MATEN: ONGEVEER 7 X 7 CM. VERVOLGENS ZIJN MOGELIJKHEDEN:

A. BRENGT DE ADRESSEN HEX. 0000-3FFF, A000-AFFF, E000-EFFF, B010-B3FF, B410-B7FF, B810-BBFF EN BC10-BFFF BUITEN DE BUS.
B. GEINVERTEERDE R/W T.B.V. DATABUFFER (ZIE OOK AN 5-83 BLZ 40).
C. OMSCHAKELEN NAAR BBC MODE, ZONDER BEPERKINGEN.
D. LAGE KOSTEN (ONGEVEER F 30,--)

MAAK MAAR EENS EEN MEMORY MAP ZOALS UW GEHEUGEN NU INGEDEELD IS EN MAAK ER DAN EEN ZOALS ONDER PUNT A VOORGESTELD WORDT. HET VERSCHIL ZAL U SNEL DUIDELIJK WORDEN.
VOOR PUNT B VERWIJS IK U NAAR EEN VORIGE UITGAVE VAN ACORN NIEUWS.
PUNT C VEREIST WAT MEER UITLEG. DAAR DE SCHAKELING GEBRUIKT MAAKT VAN ALLE INGANGEN VAN IC5 ONTSTOND DE MOGELIJKHEID OM DE BUFFERS AUTOMATISCH MEE TE LATEN SCHAKELEN. WANNEER BIJV. IN BBC MODE HET VIDEODEEL GEADRESSEERD WORDT OP HEX 4000-5FFF, DAN IS DE DATABUFFER IN DIT GEBIED NIET ENABLE, ZODAT DE 16K KAART (HEX 4000-7FFF) NIET UITGESCHAKELD HOEFT TE WORDEN. OP DEZELFDE WIJZE SCHAKELT DE SCHAKELKAART UIT.

VERDER BEVINDT ZICH OP HET PRINTJE NOG EEN CONNECTOR VOOR DE BBC ROM EN DE MOGELIJKHEID OM IN BBC MODE HET ATOM VIDEODEEL UIT TE SCHAKELEN. DIT ADRES-GEBIED KOMT DAN VANZELF OP DE BUS.
IK HAD DEZE MOGELIJKHEDEN KORT AANGETIPT IN MIJN ARTIKEL OVER DE 2E 16K KAART, DOCH DOOR EEN KLEINE WIJZIGING IN DE TEKST KOMT DIT ER NIET UIT. GELIEVE IN DAT ARTIKEL (AN 5-83 BLZ 41) NIET VERDER TE LEZEN DAN "NOG EEN MOGELIJKHEID".

INDIEN U INMIDDELS EEN MEMORYMAP HEBT GETEKEND, ZOALS VERMELD ONDER PUNT A, DAN ZULT U ZIEN DAT ALLES, BEHALVE DE EERSTE 16 BYTES VAN DE I/O'S, NAAR BUITEN WORDT GEBRACHT. ALLE VIA'S EN PIA'S KUNNEN NU NETJES OP DE BUS AANGESLOTEN WORDEN ZONDER BOSSEN DRAAD UIT DE KAST !! HEBT U AAN 16 BYTES INTERN NIET GENOEG, DAN KAN DAT OP SIMPELE WIJZE VERANDERD WORDEN IN 32, 64 OF NOG MEER. DIT GEBEURT DAN NATUURLIJK STEEDS VIER KEER !! OVERIGENS BLIJFT ONVERKORT DE MOGELIJKHEID BESTAAN OM DEZE STUKJES NOG WEER OP TE DELEN IN BLOKJES VAN VIER MET BIJV. 74LS139.

KORT SAMENGEVAT ONTSTAAT DE MOGELIJKHEID OM OP WEL ZEER SIMPELE WIJZE BBC ROM TE KUNNEN GEBRUIKEN EN DE ATOM MOGELIJKHEDEN (NOG !!) VERDER TE VERRUIMEN.
EEN PROTOTYPE WERKT BIJ MIJ PROBLEEMLOOS ALHOEWEL IK NOG NIET MET EEN BBC ROM HEB KUNNEN TESTEN. ALLE ATOM MOGELIJKHEDEN DOEN HET ECHTER PRIMA.
MOCHT U BELANGSTELLING HEBBEN, LAAT DIT DAN VIA UW REGIONALE COORDINATOR WETEN. BIJ VOORKEUR SCHRIFTELIJK.

BLIJKT DEZE BELANGSTELLING VOLDOENDE, DAN ZAL IK PROBEREN HIERVAN EEN COMPLEET ONTWERP MET PRINT TE MAKEN EN DIT T.Z.T. IN ACORN NIEUWS TE PUBLICEREN.
VOOR DIT ONTWERP DIENT U WEL IN HET BEZIT TE ZIJN VAN EEN TWEEDE RAM KAART. BOVENDIEN MOETEN RAM EN SCHAKELKAART BUITEN DE KAST, MAAR DAT WAS U TOCH AL VAN PLAN !!

Het hier beschreven programma werd door ons ontwikkeld, omdat we beiden moeilijkheden ondervonden met de 16-k kaart. Na het lezen van een artikel in een BYTE van enkele jaren geleden besloten we toen om zelf ook maar eens een goed test-programma voor RAM's te schrijven. Wat is namelijk het geval? Sommige geheugen IC's vertonen een moeilijk te localiseren fout: de zogenaamde patroonsgevoeligheid. Wat houdt dit in. Als er op een bepaalde locatie een specifiek bitpatroon staat dan kan dit de inhoud van andere locatie's beinvloeden bij operati's die we op dit geheugen IC willen uitvoeren.

Om dit te testen vullen we het geheugen met opeenvolgende waarden. (bv: ?#4000=#1C;?#4001=#1D enz.). Daarna zetten we de inhoud van de hoogste te testen plaats op die van de laagste te testen plaats enz. Als het bereik van bv. #4000 tot #5000 is, dan houdt dit in: T=?#4000;?#4000=?#5000;?#5000=T;T=?#4001;?#4001=?#4FFF=T enz. Waarna we gaan testen of de goede waarden in de verschillende plaatsen staan. Dit kunnen we dan voor alle beginwaaarden herhalen.

Als extra hebben we ook nog een test toegevoegd, die elk afzonderlijk bit in het te testen gebied bekijkt op zijn juiste werking. Verder wordt ook nog getest of een geheugen gedeelte dubbel geadresseerd wordt. (BB0 t/m BB12).

```
   10 REM PATTERN TEST
   20 DIM II3,SS3,TT6,DD0,BB12,L-1
   30 F.I=0TO29;III=L;N.I
   40 Z=#80;O=Z+4;B=Z+6;C=Z+8;Q=Z+10
   50 P.$21;GOS.z
   80 GOS.z;P.$6
   90 P.$12"MEMORY PATTERN SENSITIVITY TEST"'
  100 P."PLEASE GIVE RANGE"'
  110 IN."FROM "X;Z!0=X
  120 IN."TO   "Y;Z!2=Y
  130 P."TEST1:BIT TEST"';LINKBB0
  140 P."=================================="
  150 A=ABSRND%256
  160 P."TEST2:PATTERN TEST"'
  170 P."=================================="
  180 P."STARTING VALUE IS:"&A'
  190 P."INITIALISING"';LINKII0
  200 P."SWAPPING"';LINKSS0
  210 P."TESTING"';LINKTT0
  220 END
  230zP=L
  240[
  250:II0 JSRDD0
  260:II1 STA(O),Y;CLC;ADC@1
  270 INCO;BNEII3;INCO+1
  280:II3 LDXO+1;CPXB+1;BNEII1
  290 LDXO;CPXB;BNEII1
  300:II2 STA(O),Y;RTS
  310:SS0 JSRDD0
  320:SS1 LDA(O),Y;PHA;LDA(B),Y;STA(O),Y;PLA;STA(B),Y
  330 INCO;BNESS2;INCO+1
  340:SS2 DECB;LDAB;CMP@#FF;BNESS3;DECB+1
  350:SS3 LDAB+1;CMPO+1;BNESS1;LDAB;CMPO;BNESS1;RTS
  360:TT0 JSRDD0
  370:TT2 CMP(B),Y;BEQTT3
```

```
380:TT1 JSRTT5
390:TT3 CLC;ADC@1
400 LDXB;BNETT4;DECB+1
410:TT4 DECB
420 LDXB+1;CPXD+1;BNETT2
430 LDXB;CPXD;BNETT2
440 CMP(B),Y;BEQTT6
450 JSRTT5
460:TT6 RTS
470:TT5 PHA;JSR#F7D1
480←;$P="ERROR AT ";P=P+L.P
490↑LDX@B;JSR#F7F1;JSR#F7D1
500←;$P="MUST BE: ";P=P+L.P
510↑NOP;PLA;PHA;JSR#F802;JSR#FFED;PLA;RTS
520:DD0
530 LDYZ+0;STYD;LDYZ+1;STYD+1
540 LDYZ+2;STYB;LDYZ+3;STYB+1
550 LDY@0;RTS
560:BB0 JSR DD0;JSR BB4;JSR BB6
570:BB1 JSR BB10;BNE BB3;LDA @#FF;STA (C),Y;JSR BB7;BCS BB1
580 JSR BB4;LDX B;STX C;LDX B+1;STX C+1
590:BB2 JSR BB10;BNE BB3;LDA @#FF;STA (C),Y;JSR BB9;BCS BB2
600 JSR #F7D1;←;$P="RAM TEST1 SUCCESFUL;NO ERRORS  ";P=P+L.P;↑
610 NOP;JSR #FFED;RTS
620:BB3 JSR #F7D1;←;$P="ERROR AT:";P=P+L.P;↑
630 LDX @C;JSR #F7F1;JSR #FFED;RTS
640:BB4 JSR BB6;LDY @0
650:BB5 LDA @0;STA (C),Y;JSR BB7;BCS BB5;RTS
660:BB6 LDX D;STX C;LDX D+1;STX C+1;RTS
670:BB7 INC C;BNE BB8;INC C+1
680:BB8 SEC;LDA B;SBC C;LDA B+1;SBC C+1;RTS
690:BB9 SEC;LDA C;SBC @1;STA C;LDA C+1;SBC @0;STA C+1;SEC
700 LDA C;SBC D;LDA C+1;SBC D+1;RTS
710:BB10 LDA @1;STA Q;LDY @0;LDA (C),Y;BNE BB11;LDX @8
720:BB11 LDA Q;STA (C),Y;CMP (C),Y;BNE BB12;ASL Q;DEX;BNE BB11
730:BB12 RTS
740←
750 RETURN
```

Toelichting:

40: Zero-page loktie's
110: Ondergrens inlezen.
120: Bovengrens inlezen.
250: Vul geheugen met opeen volgende waarden.
310: Swap de inhoud.
360: Test of de inhoud correct is.
520: Initialiseer de grenzen.
560: Bittest (zie ook ELEKTUUR april 1982.)

TEN AANZIEN VAN HET GEBRUIK VAN HET PROGRAMMA IS HET BELANGRIJK OM TE VERMELDEN DAT DE 'RANGE' DIE OPGEGEVEN MOET WORDEN UIT EEN oneven AANTAL BYTES MOET BESTAAN. ALS EEN EVEN AANTAL BYTES WORDT OPGEGEVEN, STOPT HET PROGRAMMA NIET ALS HET MET 'SWAPPING' BEZIG IS, EN VEEGT ALLES WEG WAT HET DAARBIJ TEGENKOMT, INCLUSIEF ZICHZELF. EEN RANGE WORDT DUS NIET B.V. #4000-#57FF OPGEGEVEN, MAAR ALS #4000-#57FE OF #4001-#57FF.

Het samenstellen van een zgn. Shape-tabel kan op twee manieren:
- in zgn. Base Two
- in zgn. Base Four

Base Two wil zeggen dat de definitie gemaakt wordt met behulp van slechts 2 cijfers, nl. 0 en 1.n een programma dient VOOR elke rij nullen en eenen een geinverteerde b (lees: kleine b) geplaatst te worden.

Base Four maakt gebruik van 4 cijfers nl. 0, 1, 2 en 3. In het programma dient nu VOOR elke rij cijfers een geinverteerde f (lees: kleine f) te worden getypt.

Base Four neemt minder geheugenruimte in, in het Basic-programma; maar is niet erg overzichtelijk. Maar daarover straks. Allereerst Base Two.

BASE TWO

Wanneer je het bijgeleverde programma erbij neemt kun je de Shapetabel vergelijken met hetgeen ik hieronder neerschrijf.

Bij het maken van een Shape-tekening kan het beste worden uitgegaan van een tekening op ruitjespapier. Breng op zo'n stuk papier een kader aan dat 8 x 8 ruitjes omvat zoals hieronder weergegeven en vergelijk de tekening met de Shape-tabel uit het programma. Overal waar een hokje zwart gemaakt is staat in de tabel een 1. Dit hokje of die 1 wordt op het scherm een witte pixel!

Om er voor te zorgen dat de figuurtjes straks ook netjes over het beeldscherm bewegen zonder SPOREN achter te laten dienen deze figuurtjes op het juiste moment te worden uitgeveegd of juist op het scherm geplaatst. Dit kan m.b.v. de instructie INV. Vanaf regel 200 kun je controleren hoe dit in z'n werk gaat. Het commande MOV. spreekt voor zich.

BASE FOUR

Wanneer je de binaire code van de cijfers 0, 1, 2 en 3 op een rijtje zet kun je zien wat deze@cijfers in een Shape-tabel betekenen:

0	00
1	01
2	10
3	11

Wanneer je dus de Shape-tabel uit het programma wilt omzetten naar Base Four moet je steeds 2 Base Two cijfers vervangen door 1 Base Four cijfer! dus:

b11000000 wordt f3000
b01100000 wordt f1200
b00111000 wordt f0320

enz,etc,usw,...

Trouwens het programma laat het ruimtescheepje dat overigens uit twee aan elkaar gekoppelde SHAPE's bestaat ook schuin over het scherm bewegen.

Het programma is beslist niet perfect maar geeft wel een leuke indruk van de mogelijkheden van het SHAPE-commando. De kleine assembler-routine aan het begin van het programma staat min of meer letterlijk in de Engelse handleiding op pag. 116 en deze verzorgt het geluid.

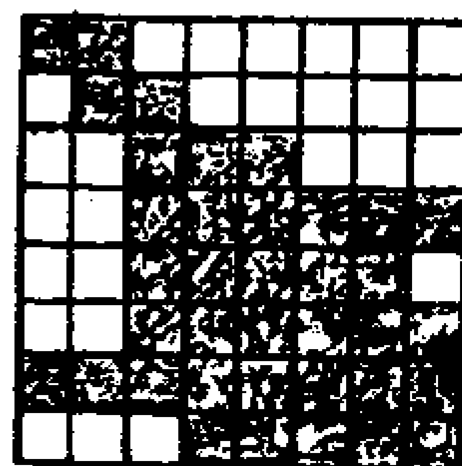
LET OP: Wanneer je geen 64k of club geheugenkaart hebt moeten de volgende adressen worden veranderd:

rgl. 20 - P=#7000 kan worden P=#3000 o.i.d.

rgl. 55 - FR.=#7100 kan worden FR.=#3100 (in ieder geval #100 hoger dan het adres op rgl. 20)

rgl. 5 heeft betrekking op het voorschakelen van de SUPERBASIC toolbox en zal voor elke gebruiker verschillend zijn.

```
   1REM      shape
   2REM hans heuts
   5P.$21;?#BFFF=4;LINK44600;P.$6
  10DIML2,NN2;NN0=-1;NN1=-1;NN2=-1
  20R=#B002;P.$21;F.Q=0TO1;P=#7000
  25↑:NN0 LDY@0
  30:NN1 LDAL;STAR;AND@#48;ADC@#38;ASLA;ASLA;ROLL+2;ROLL+1;ROLL
  40INY;CPY@#FF;BNENN1;:NN2RTS;←
  50N.;P.$6;G=0
  55FR.=#7100;DIME40,F40;F.Q=0TO40;E?Q=-20;F?Q=-20;N.
  60DE.M,b11000000,b01100000
  70DE.' b00111000,b00111111
  80DE.' b00111110,b00111111
  90DE.' b11111111,b00011111
 100DE.L,b00000000,b00000000
 110DE.' b00000000,b11110000
 120DE.' b10011100,b11111111
 130DE.' b11111111,b11111000
 140DE.P,b00011000,b01100110
 150DE.' b10100101,b00011000
 160DE.' b00011000,b00011000
 170DE.' b00100100,b00100100
 180CLEAR4
 190F.N=0TO19;Y=A.R.%50+90
 200INV.M(0,Y),L(8,Y)
 210T=1;C=0;D=17;O=A.R.%200+20
 215S=A.R.%6+3
 220F.X=0TO255S.S
 230IFX)O A.X(=O+S;GOS.p;E?G=X;F?G=Y;G=G+1
 235IFX)O+30INV.P(E?(V-1),F?(V-1));GOS.e;INV.P(E?(V-1),F?(V-1))
 236IFX)O+30X=255;N.;G.300
 240M.M(X,Y,X+S,Y+T),L(X+8,Y,X+S+8,Y+T);LI.NN0
 250IFX=100D=10;C=0
 260Y=Y+T
 270C=C+1;IFC=D;C=0;T=-T;D=D-2
 275GOS.p;?#14=0
 280N.
 290INV.M'(X,Y),L'(X+8,Y)
 300N.;E.
 310bF.Q=0TO2;F.B=80TO40S.-10;BE.B,1;GOS.r;N.;N.;R.
 320rF.W=0TO2;LI.NN0;N.;R.
 330pIFG=0R.
 335G=G-1
 340F.V=0TOG;H=S+3
 345IFF?V(=7ORF?V)=200F?V=7;INV.P(E?V,F?V);N.
 350INV.P(E?V,F?V)
 355M.P(E?V,F?V,E?V,F?V-H)
 360INV.P(E?V,F?V-H);F?V=F?V-H
 370N.;G=G+1;R.
 380eINV.M'(X,Y),L'(X+8,Y);F.Q=3TO22;F.U=0TO1
 385H=Q*2;I=Q+5;J=Q*3;K=Q+11
 390GOS.u;PLOT10,-H,I;GOS.u;PLOT10,-K,-H
 400GOS.u;PLOT10,J,-K;GOS.u;PLOT10,I,I
 410GOS.u;PLOT10,0,I;GOS.u;PLOT10,-H,0
 420GOS.u;PLOT10,0,-J;GOS.u;PLOT10,H,0;N.;N.;R.
 430uPLOT4,X,Y;R.
```



Vrijwel iedereen heeft iemand in de naaste omgeving die wel eens vraagt: "Wil je me even overhoren?", want helaas is het nog steeds nodig om bepaalde feiten 'uit 't hoofd' te leren (memoriseren noemen de vakmensen dat).

Dat overhoren is een taak die de computer op het lijf is geschreven. Allerlei feitenmateriaal kan worden overhoord: woorden in een andere taal, formules, topografische gegevens, standaardoplossingen, enz. enz.

Voer het volgende programma in.

```
 1Ø P.Ø12"overhoorprogramma"''
 2Ø REM JBDEKKER
 3Ø IN."OM HOEVEEL VRAGEN GAAT HET "X; DIM AAX, BBX,CCX
 4Ø F.Y=1 TO X; DIMQ64; AAY=Q; DIMR64; BBY=R;
    DIMS64; CCY=S; NEXT Y
 5Ø F.Y=1 TO X; @=Ø; P."VOER VRAAG "Y" IN "'; IN.ØAAY
 6Ø @=Ø: P."VOER ANTWOORD "Y" IN "';IN.ØBBY; P.Ø12;N.Y
 7Ø F.Y=1 TO X;P.ØAAY'; IN.ØCCY
 8Ø ?#E1=Ø; GOSUB A.
 9Ø LI.#FFE3; P.Ø12; NEXT Y; G.7Ø
1ØØA IF ØBBY=ØCCY;P."GOED!"'; RETURN
11Ø P.'"NEE, HET IS: "ØBBY; RETURN
```

Wanneer de computer print: "voer vraag x in", dan is het uiteraard niet persé noodzakelijk een echte vraag in te voeren, als degene die wordt overhoord maar weet waar het om gaat.

Er kan (b.v. bij het memoriseren van woorden) worden volstaan met het invoeren van het nederlandse woord bij de vraag en het woord in een andere taal bij het antwoord (of omgekeerd).

U zult zien: vooral kinderen vinden deze manier van overhoren prachtig.
Ook is het iedereen eenvoudig bij te brengen hoe, met behulp van het programma, het overhoormateriaal ingevoerd moet worden.
En ook dat heeft al een zeker leerrendement.

Wie vindt dat er te veel op knoppen moet worden gedrukt, kan regel 9Ø wijzigen:

```
9Ø FOR N=Ø TO 12Ø;WAIT;N.N;P.Ø12;N.Y; G.7Ø
```

Er zijn uiteraard ook mensen die het niet prettig vinden om met het juiste antwoord te worden geconfronteerd als een vraag foutief wordt beantwoord.

Om dat te voorkomen, kan het programma als volgt worden gewijzigd:

De volgende regels toevoegen:

```
 45P."HOE VAAK MOET ELKE VRAAG WORDEN "
 46P."HERHAALD VOORDAT IK HET JUISTE  "
 47IN."ANTWOORD GEEF"H;P.Ø12
12Ø P."NEE, NIET GOED"';G.8Ø
```

De volgende regels wijzigen:

```
 7Ø FOR Y=1 TO X; P.Ø12; V=Ø
 8Ø P.ØAAY; IN.ØCCY; P.Ø12; GOSUB  A
11Ø V=V+1; IF V=(H+1); P."NEE, HET IS: "ØBBY; RETURN
```

Ik hoop dat dit programma mag bijdragen tot gemakkelijker "instampen" van feitenkennis!

J.B.Dekker

Acorn computerclub
regio Overijssel/Gelderland

Vanaf dit moment is er weer een reden minder om van de ATOM over te schakelen op de BBC. Met onderstaande hard- en software beschrijving bent u in staat een SOUNDBOARD te bouwen voor zo'n 80 gulden welke dezelfde mogelijkheden biedt als de meeste (modernere ?) homecomputers met standaard geluidseffecten.
De soundboard is opgebouwd rond de AY-3-8910 van General Instruments, dit is een muziekchip welke reeds in de meeste computers als geluidsgenerator is toegepast. Deze chip heeft (behalve een externe versterkerschakeling) geen extra componenten nodig, alle besturingen gebeuren via data-adreslijnen.

In dit artikel vindt u:

- een hardware beschrijving van de SOUNDBOARD
- een beschrijving van de PSG (d.i. de AY-3-8910, PSG staat voor Programmable Sound Generator)
- een beschrijving van de PSG-functies
- een softwarebeschrijving in twee delen, t.w. een aantal MC-routines (welke bedoeld zijn voor gebruik binnen Basic programma's) en een FORTH-applicatie. Forth namelijk leent zich erg goed voor dergelijke toepassingen, vanwege hoge snelheid en de mogelijkheid om eenvoudig nieuwe instructies te definiëren. De Forth-applicatie is speciaal bedoeld om te onderzoeken wat zoal de mogelijkheden zijn van de PSG.
- een aantal voorbeelden

Wat is in elk geval nodig voor deze SOUNDBOARD:

- de AY-3-8910 en enkele componenten voor een versterkerschakeling, daarnaast zaken zoals connector(s), een luidsprekertje ed. Voor een componentenlijst zie ook de hardware beschrijving.
- de 6522 VIA met minimaal 1 1/2 poort beschikbaar, de parallelpoorten van de VIA worden gebruikt als data-adreslijnen.
- enige soldeerervaring
- eventueel de FORTH cassette, maar is niet strikt noodzakelijk

Wat zijn de mogelijkheden:

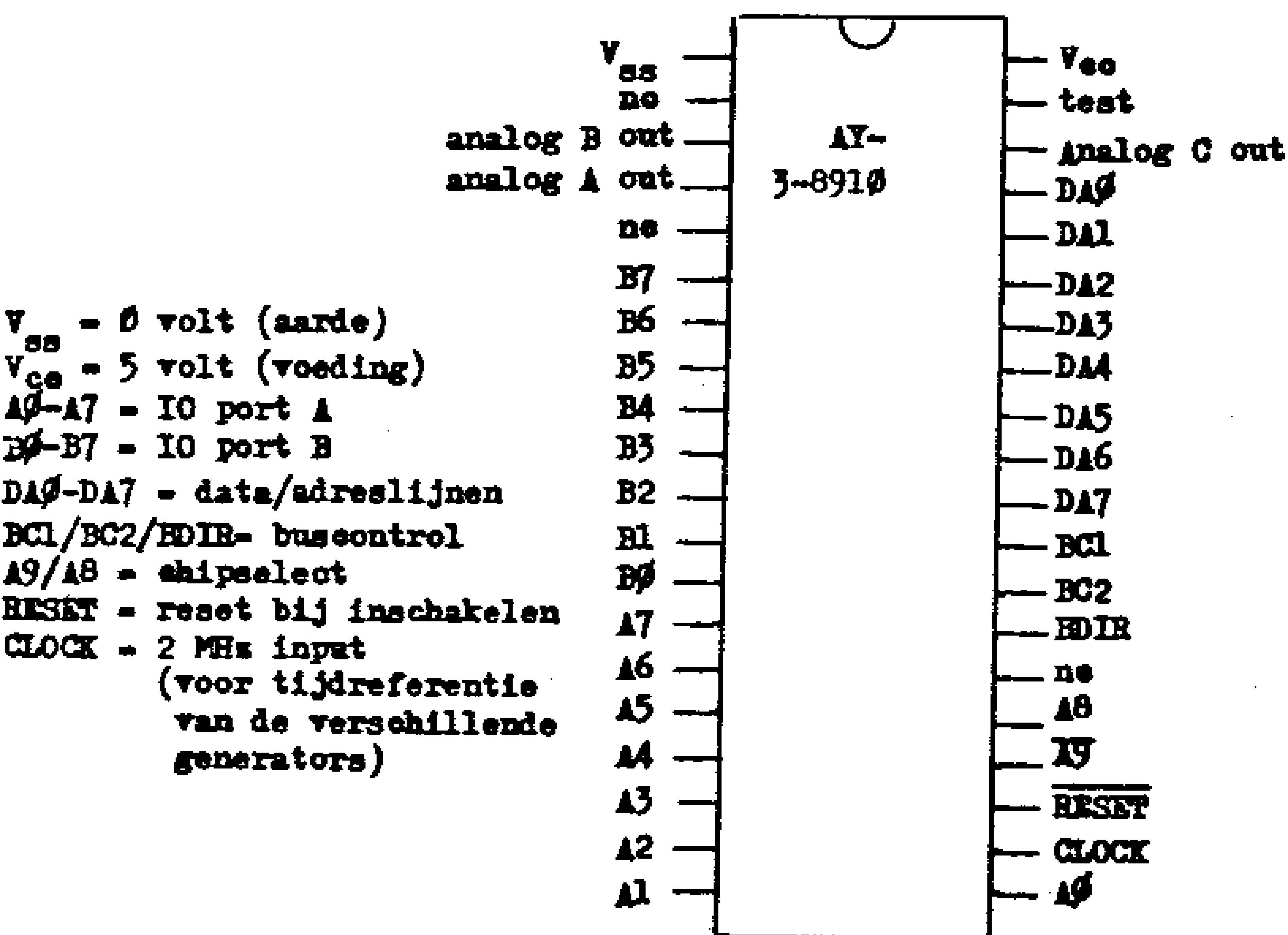
- 3 onafhankelijke, volledig programmeerbare, geluidskanalen (zowel voor klank als ruiseffecten), elk kanaal is programmeerbaar in frequentie (toonhoogte, voor ruis is dat de ruiskleur) en amplitude. De amplitude (volume) en toonduur zijn ook middels de envelope generator te besturen.
- envelope generator, voor allerlei tremolo- en aanslageffecten
- twee extra 8 bits parallel poorten, ter vervanging van de gebruikte VIA poorten of voor aansluiting van een tweede PSG of b.v. een interface naar een toetsenbord.

Voor wat betreft de geluidsmogelijkheden zelf, deze zijn alleen begrensd door uw eigen fantasie.

In het schema ontdekt men de volgende onderdelen:

- bus control decoder: bepaalt of het gaat om data of een adres (aangeboden op DA7-DAØ)
- register adres latch/decoder: decodeert het registernummer (adres)
- bi-directionele buffers: hierin staat de inhoud welke in een bepaald register moet worden weggeschreven (bepaald door register adres latch/decoder) of data welke uit een register zijn gelezen.
- register array: d.i. het blok van 16 registers (variërend van 4 tot 8 bits) waarmee alle outputs en I/O poorten worden bestuurd, de afzonderlijke functies van deze registers komen nog aan de orde.
- IO port A en B: dat zijn de beide parallel IO poorten (2 x 8 bits).
- Noise generator: ruisgenerator welke naar keuze met een of meer van de oscillatoruitgangen gemixed kan worden.
- Oscillator A, B en C: dat zijn de eigenlijke toongenerators
- Amplitude control: regelt de amplitude van de afzonderlijke uitgangssignalen (voor elke uitgang A, B en C apart).
- Envelope generator: de amplitude controller kan worden bestuurd door de envelope generator, deze geeft een signaal af welke de uiteindelijke contour (= amplitudeverloop) van het uitgangssignaal bepaalt.
- Mixer A, B en C: hierin worden de signalen van oscillator A, B en C en de ruisgnerator gemengd.
- D/A converters: vormen het digitale signaal om tot een analoog signaal.

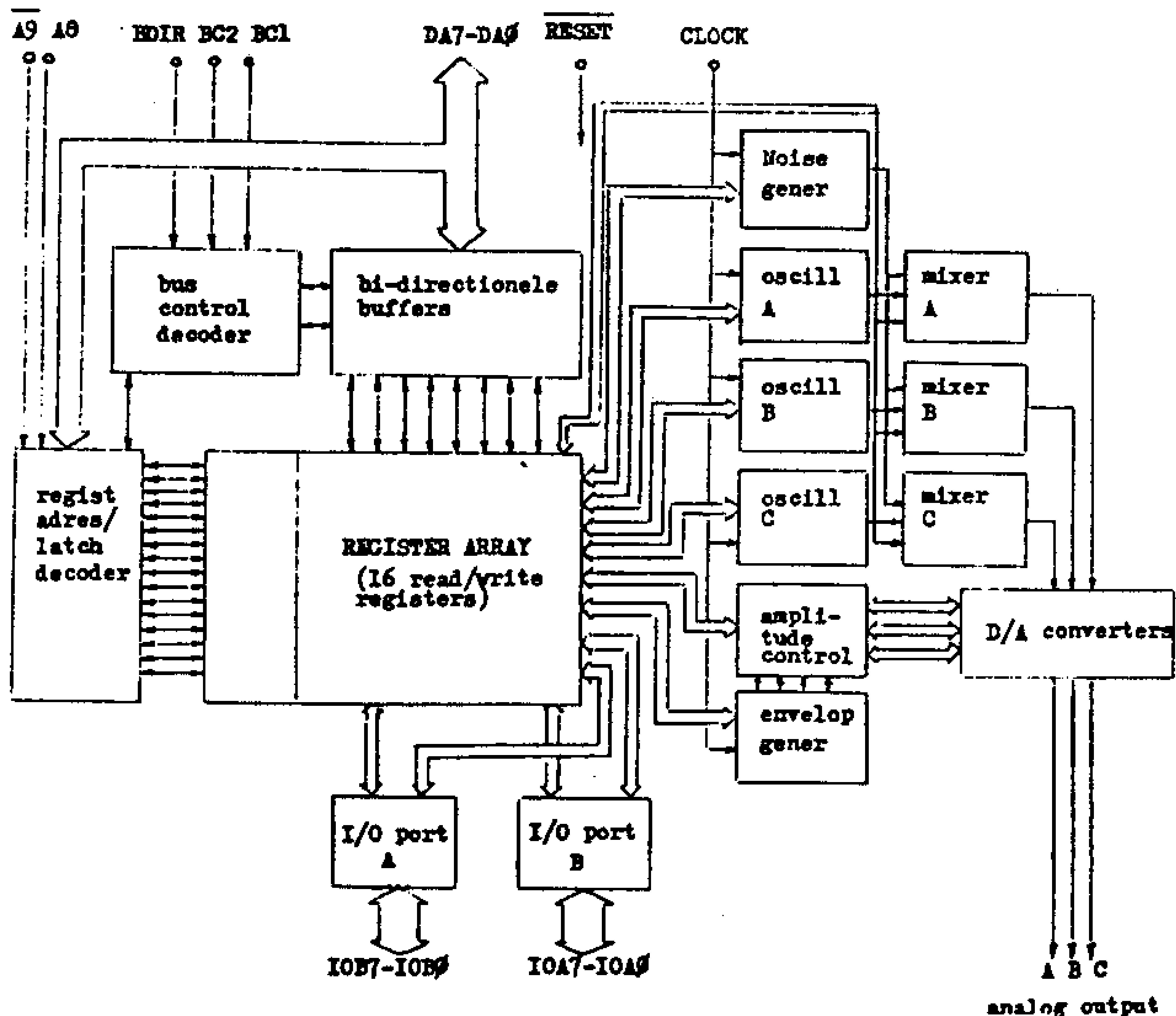
Hieronder ziet u de pin configuratie van de AY-3-891Ø:



De hiernavolgende afbeeldingen geven de aansluitschema's weer van de soundboard aan de atom. Men kan dit schema eenvoudig nabouwen op een stukje experimenteerprint, uiteindelijk zijn er slechts weinig externe componenten nodig en zal het meeste soldeerwerk bestaan uit het maken van draadbruggen.

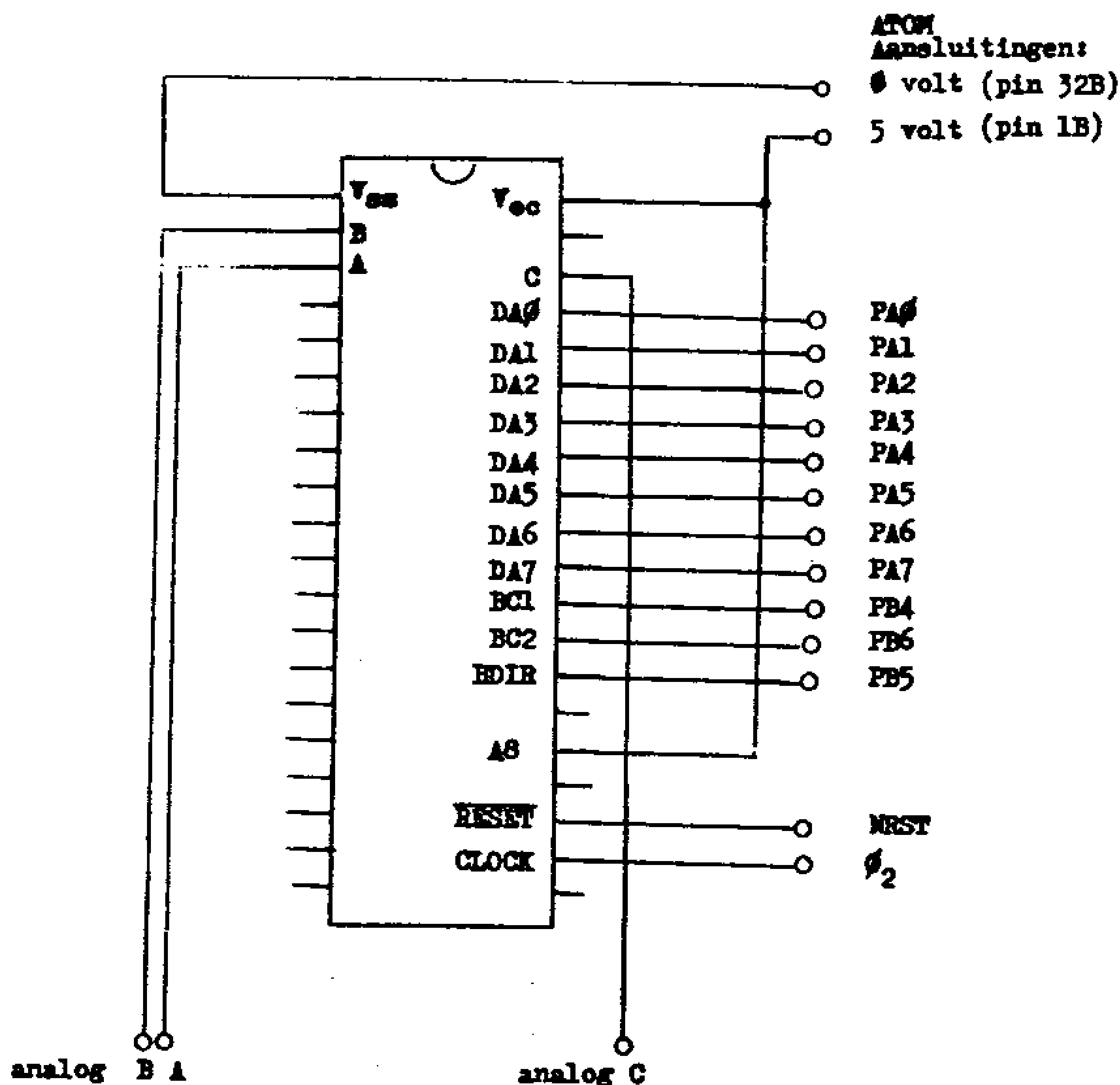
Onderstaande schema geeft de opbouw weer van de AY-3-8910, het ligt niet in mijn bedoeling dit schema uitputtend te beschrijven, echter enkele belangrijke zaken zijn aangegeven.

Feitelijk draait alles om een zestiental registers waarmee alle besturingen geregeld worden (zoals frequentie-amplitude-input/output ed).
Het vullen van deze registers gebeurt via de datalijnen DA7-DAØ, deze lijnen worden tevens gebruikt voor de adressering (om 'pootjes' te besparen).
Om aan te geven of DA7-DAØ als data- of als adreslijnen functioneren zijn de aansluitingen BDIR, BC1 en BC2 (respectievelijk Busdirection, buscontrol 1 en buscontrol 2) nodig. Hierover later meer.



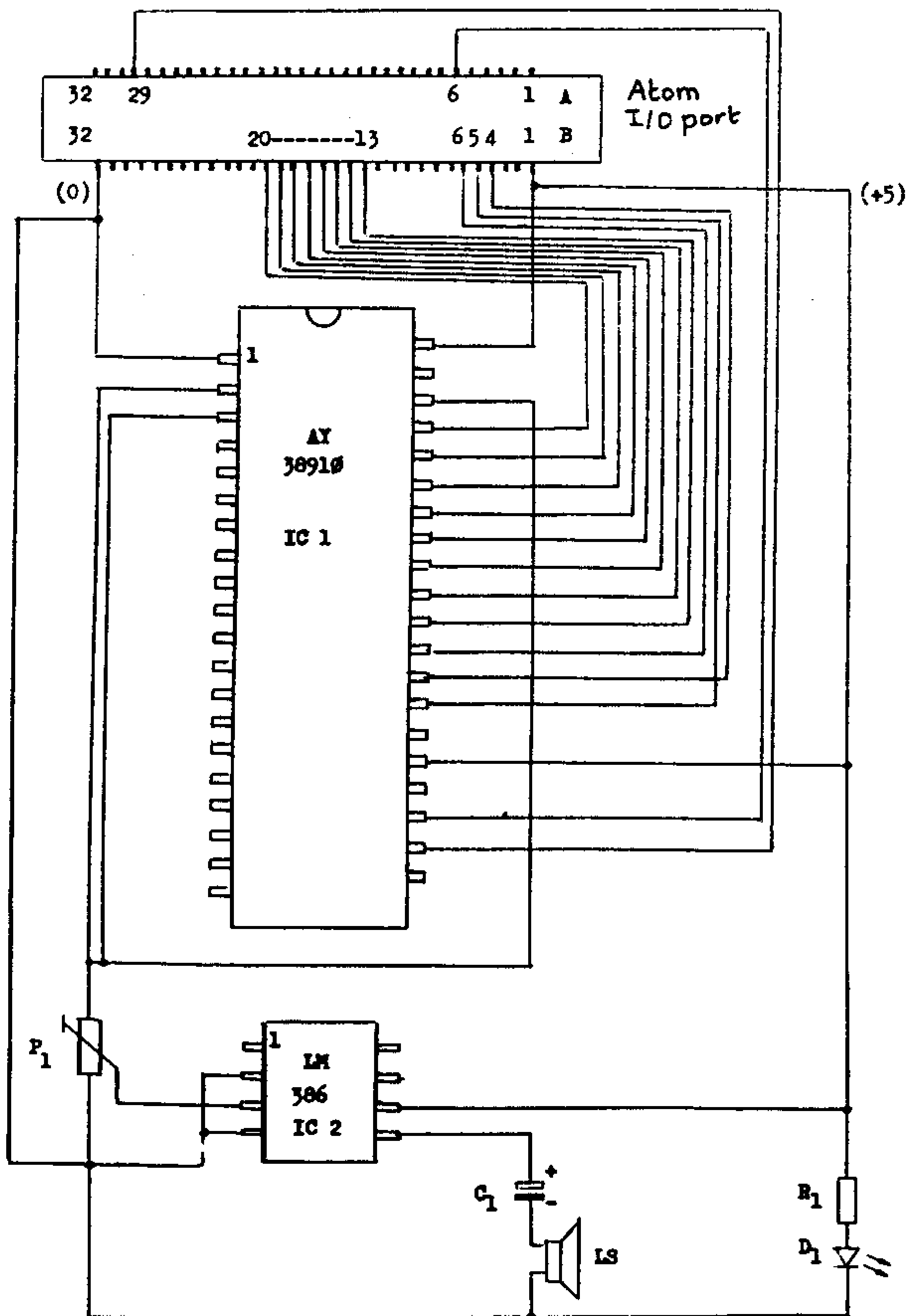
De lijnen DA7-DAØ zijn bidirectioneel, d.w.z. ze kunnen in twee richtingen gebruikt worden (nl. om data weg te schrijven naar de registers en om data uit de registers terug te lezen).
De aansluitingen A9 en A8 zijn chip-select lijnen, voor het al dan niet inschakelen van de AY-3-8910.

Aansluitingen van de PSG met de Atom bus extension



De lijnen DA7-DAØ van de PSG zijn aangesloten aan de lijnen van de VIA poort A: PAØ-PA7.
Daarnaast zijn van de VIA poort B een drietal lijnen gebruikt voor de aansluitingen BC1, BC2 en BDIR (d.i. voor de besturing van de lijnen DA7-DAØ).
De reden waarom hiervoor de aansluitingen PB4 t/m PB6 zijn gebruikt ligt hem in het feit dat deze op mijn Atom nog vrij waren. De aansluitingen PBØ t/m PB3 heb ik nl. reeds in gebruik (voor epromkaart en 64K ram kaart). Vanzelfsprekend kunt u de BC1, BC2 en BDIR van de PSG aansluiten aan PBØ - PB2, echter hier zal dan in de software rekening mee gehouden moeten worden (zie software-beschrijving).

Tenslotte dan het totale aansluitschema van de SOUNDBOARD (inclusief versterkerschakeling):



AANSLUITSCHEMA SOUNDBOARD

Componentenlijst:

IC1 : AY-3-8910 van General Instruments, deze is in elk geval te krijgen bij de Boer electronica in Eindhoven.
IC2 : LM386 (versterker IC)
P_1 : instelpotmeter 10 k
C_1 : 470 µF elko
R_1 : weerstand 680 ohm
D_1 : led naar keuse
LS : luidspreker (8 ohm of meer)
diversen: stukje experimenteerprint (bv. 10 x 10 cm)
64 pens connector voor de atom
eventueel nog een 64 pens connector en een 64 pens contraconnector
IC voetjes

Enkele opmerkingen:

* zorg ervoor dat de aansluitingen op de print precies kloppen, controleer de nummering van de I/O bus van de atom (Atom bus extension), raadpleeg zonodig de schema's van de Atom en van de 6522 VIA.
  Let op dat het verschil in A- en B-aansluitingen van de connector essentieel is.
* gebruik IC-voetjes, monteer alles, controleer de aansluitingen, plaats de IC's pas dan, wanneer u echt zeker bent van de print
* de aansluiting van de LED (en dus ook de weerstand R_1) kan vervallen, deze is namelijk niet functioneel, maar geeft een aardig gezicht indien het geheel in een kast wordt ingebouwd.
* de instelpotmeter moet nog worden afgeregeld, indien de testprogramma's goed werken, regel de potmeter dan af op het gehoor. Het signaal moet hard genoeg klinken maar mag niet vervormd zijn.
* Gebruik voor de aansluiting van de print met de atom een 64 pens connector, het geheel kan dan zo in de Atom gestoken worden.
  Indien u het geheel in een kastje wilt bouwen, monteer dan op de print een gewone 64 pens connector (dus zo een welke direct in de Atom past) en monteer een stuk flatkabel (20-25 aderig) met aan een zijde een gewone connector en aan de andere zijde een contraconnector (dus zo een als welke in de Atom is gemonteerd). Het geheel blijft op deze wijze demontabel en er lopen ook geen losse verbindingen naar de print.

2. De PSG registers.

Zoals gezegd, draait alles om zestien registers. Elk register heeft zijn eigen functie(s), zoals in onderstaande tabel aangegeven:

Registernummer	Functie	Aantal bits
R0	fijnregeling toonhoogte kanaal A	8
R1	grofregeling toonhoogte kanaal A	4
R2	fijnregeling toonhoogte kanaal B	8
R3	grofregeling toonhoogte kanaal B	4

Registernummer	Functie	Aantal bits
R4	fijnregeling toonhoogte kanaal C	8
R5	grofregeling toonhoogte kanaal C	4
R6	grondfrequentie ruisgenerator	5
R7	enable register	8
R8	amplitude kanaal A	5
R9	amplitude kanaal B	5
R1Ø	amplitude kanaal C	5
R11	fijnregeling envelope frequentie	8
R12	grofregeling envelope frequentie	8
R13	envelope golfvorm	4
R14	I/O poort A data	8
R15	I/O poort B data	8

De frequentie van de oscillators (A, B en C) wordt dus bepaald door een 12 bits getal (respectievelijk in de registers RØ/R1, R2/R3 en R4/R5):

b11	b1Ø	b9	b8	b7	b6	b5	b4	b3	b2	b1	bØ
R1/R3/R5				RØ/R2/R4							

Dat komt dus overeen met een getal tussen Ø en 4Ø95.
De grondfrequentie van de ruisgenerator wordt bepaald door een 5 bits getal (dus van Ø tot 31), verandering van de grondfrequentie geeft een andere "ruiskleur".
Het enable-register heeft meerdere functies, namelijk:

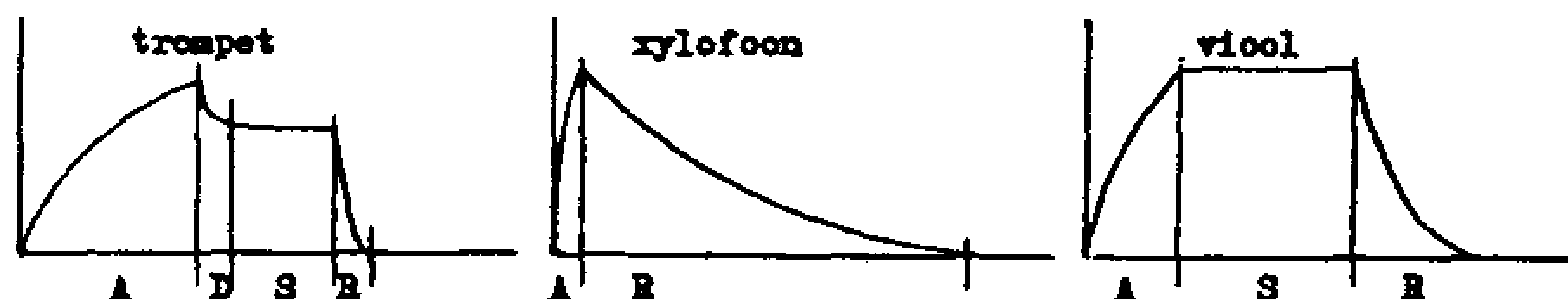
R7	bitwaarde=Ø	bitwaarde=1
bit Ø	signaal van oscillator A op mixer A	signaal niet op mixer A
bit 1	signaal van oscillator B op mixer B	signaal niet op mixer B
bit 2	signaal van oscillator C op mixer C	signaal niet op mixer C
bit 3	signaal van ruisgenerator op mixer A	signaal niet op mixer A
bit 4	signaal van ruisgenerator op mixer B	signaal niet op mixer B
bit 5	signaal van ruisgenerator op mixer C	signaal niet op mixer C
bit 6	I/O poort A als input poort	I/O poort A als output poort
bit 7	I/O poort B als input poort	I/O poort B als output poort

Het enable-register regelt dus welke signalen uiteindelijk op de uitgangen komen te staan, daarnaast functioneert hij als "Data Direction Register" voor de I/O poorten, net zoals de registers B8Ø2 en B8Ø3 van de VIA.

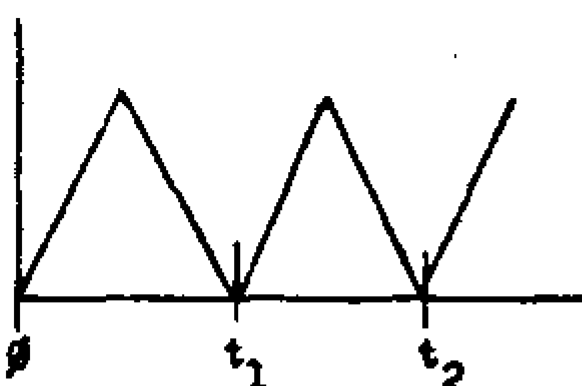
Een dergelijke curve noemt men wel de "omhullende" of de "contour" van een klank. Het karakter van een klank wordt door 3 factoren bepaald, namelijk:

- frequentie (toonhoogte)
- golfvorm (klankkleur)
- amplitudeverloop (constant, periodiek zoals b.v. tremolo of een omhullende)

Meestal duidt men de omhullende ook wel aan met "ADSR" (adsr-karakteristiek), ADSR staat voor Attack-Decay-Sustain-Release. Voor elke klank is dit een vaste karakteristiek, in electronische muziekinstrumenten (synthesizers) is de ADSR karakteristiek programmeerbaar.
Normaal gesproken bestaat deze karakteristiek uit 4 parameters, nl. de Attack in seconden, de Decay in seconden, de Sustain als een niveau-aanduiding (meestal in % van de maximale amplitude) en de Release in seconden. Het is ook mogelijk om de Sustain in twee parameters aan te geven, namelijk een sustain-niveau en een sustaintijd, dus de tijd gedurende welke een bepaald niveau wordt aangehouden.
Door de amplitude van een synthetische klank (opgewekt met een oscillator) te sturen met een omhullende, kan men bepaalde instrumenten nabootsen, ter illustratie hieronder een aantal voorbeelden van ADSR-karakteristieken van instrumenten:



Nogmaals blijkt, dat niet alle vier fasen aanwezig hoeven te zijn.
Naast ADSR-omhullenden bestaan ook repeterende omhullenden, zoals tremolo. Tremolo is een driehoekvormige, repeterende omhullende, (overigens is tremolo niet hetzelfde als vibrato, vibrato namelijk is een repeterende frequentieverandering):

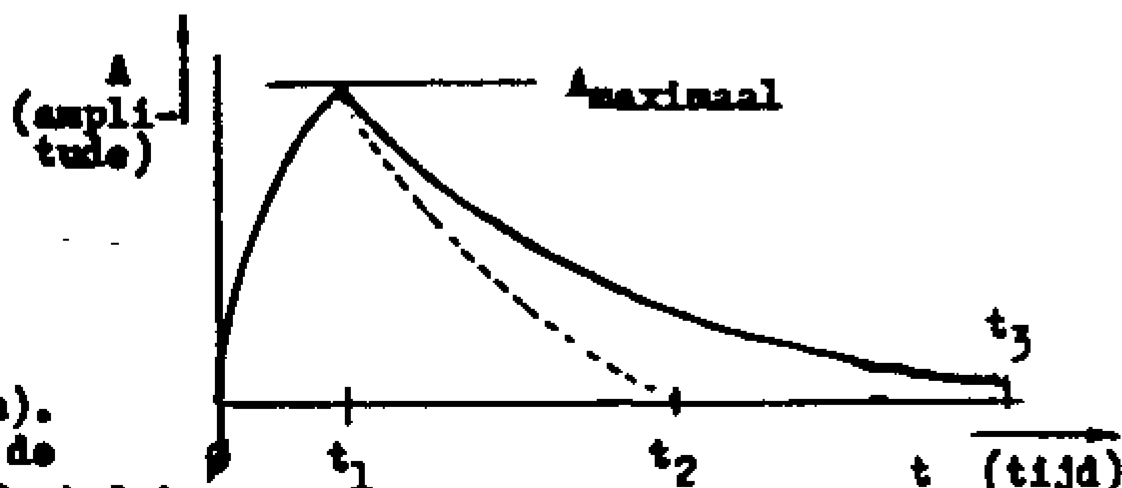


Tremolo hoeft niet persé een variatie tussen amplitude Ø en A_{max} te betekenen, het is ook heel best mogelijk een lichte variatie in de amplitude aan te brengen. Tremolo wordt daarom meestal in twee parameters aangegeven, namelijk de tremolo-intensiteit en de tremolo frequentie (d.i. het aantal variaties per seconde, ofwel de frequentie van de driehoekgolf).

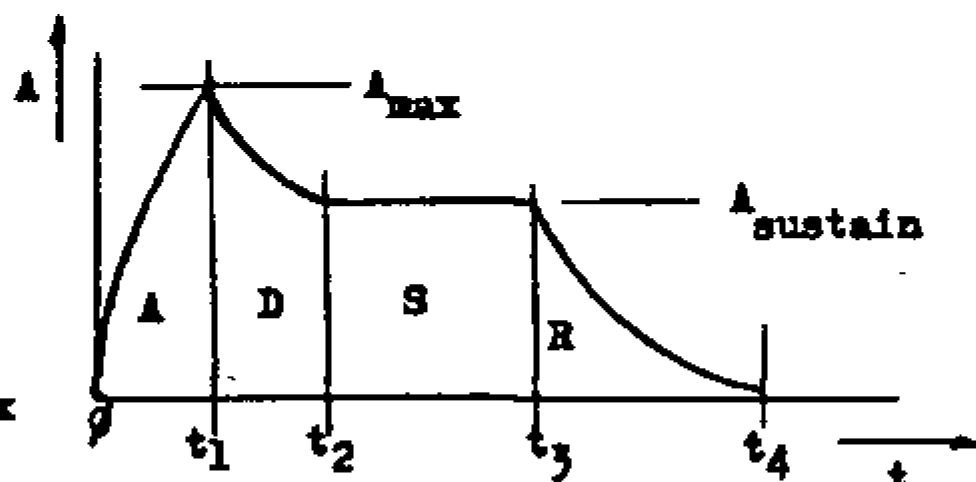
Het is overigens wel belangrijk om te realiseren dat een toonhoogte en een amplitude-karakteristiek nog geen instrument-imitatie opleveren. Zoals gezegd moet ook een golfvorm bewust gekozen worden. In synthesisers gebeurt dit door met de oscillator verschillende golfvormen op te wekken en deze in filters zo te bewerken dat de gewenste vorm ontstaat. Daarnaast spelen effecten zoals vibrato ook een grote rol.

Een golfvormkeuze is helaas niet aanwezig op de PSG, alle oscillators geven een blokgolf af.

De registers R8, R9 en R1Ø bepalen het volume van de uitgangen A, B en C. Deze registers zijn 5-bits registers, het amplitude niveau is instelbaar tussen Ø en 15 (van absolute stilte tot "hard"). Wanneer de waarde in de registers 16 is (dus als bit 5 de waarde 1 krijgt) is de amplitude van het signaal niet meer afhankelijk van de inhoud van het register, maar wordt gestuurd door de envelope generator.
Deze envelope generator kan verschillende omhullende golfvormen produceren, deze omhullende vorm is bepalend voor het karakter van de klank.
Nemen we als voorbeeld een pianoklank, indien we hiervan het amplitude-verloop bestuderen, dan valt het op dat de toename in amplitude veel sneller is dan de terugval, zoals te zien in het volgende figuur:



De amplitude neemt vrij snel toe vanaf Ø tot t_1, en bereikt de maximale amplitude. Dit veroorzaakt het effect dat een piano "aangeslagen" klinkt. De terugval duurt veel langer, namelijk van t_1 tot t_3 (doorgetrokken lijn). Deze vertraagde terugval van de amplitude veroorzaakt het effect dat de pianoklank langer "aanhoudt". Door een pedaal in te drukken kan men deze "aanhoudtijd" verkorten, en ontstaat een lijn zoals in de figuur ook is aangegeven (gestippeld).
Het blijkt dat voor alle klanken een dergelijk figuur bestaat. In het bovenstaande voorbeeld is het amplitudeverloop op te splitsen in twee fasen, namelijk een aanslagtijd en een uitsterftijd. In het algemeen deelt men het amplitudeverloop op in vier fasen, deze hebben respectievelijk de volgende benamingen:



- -Attack (aanslagtijd)
- -Decay (terugvaltijd tot sustainniveau)
- -Sustain (amplitudeniveau van een aanhoudende toon)
- -Release (terugvaltijd tot amplitude Ø)

Deze vier fasen zijn in het hiernaast afgebeelde figuur aangegeven.
De attackfase (d.i. dus de tijd tussen stilte en maximale amplitude) duurt van Ø tot t_1, de decay-fase van t_1 tot t_2, de sustainfase van t_2 tot t_3 en de release-fase van t_3 tot t_4.
Een decay- en sustainfase hoeven niet voor te komen in een amplitude-verloop, dat is bijvoorbeeld het geval in het voorbeeld van de piano. Er is bijvoorbeeld wel sprake van een decay- en sustainfase bij een orgeltoon, namelijk:

- -toets aanslaan: start attackfase
- -toets ingedrukt houden: terugval (decayfase) tot lager amplitude-
- -niveau (Sustainfase)
- -toets loslaten: releasefase

Overigens is het niet noodzakelijk dat de sustain een lager amplitude-niveau betekent, het is heel best mogelijk dat na de attack de amplitude op het maximale niveau blijft (sustainniveau= A_{max}) totdat de releasefase begint. (Dus geen decayfase).

De registers R11 en R12 zijn de beide registers welke de "Envelope period" bepalen.
R11 is de fijnregeling en R12 de grofregeling. Beide registers zijn 8 bits zodat de envelope periode dus bepaald wordt door een 16 bits getal.
Deze envelope periode is de tijdsduur van één fase van de envelope karakteristiek. (De waarde "e" in onderstaande tabel).
Register R13 bepaalt de envelope vorm, dus de vorm van de omhullende.
Dit register is 4 bits, elk bit heeft zijn eigen functie:

- bit Ø HOLD, houd de laatste amplitudewaarde aan
- bit 1 ALTERNATE, de golfvorm wordt repeterend
- bit 2 ATTACK, indien bit2=Ø een attack, indien bit2=1 een decay
- bit 3 CONTINUE, indien bit3=Ø sluit de amplitude controller af, indien bit3=1 blijft het kanaal open.

In principe zijn dus 16 verschillende envelope vormen mogelijk, in de praktijk blijken dat de volgende te zijn:

bit 3	bit 2	bit 1	bit Ø	Envelope golfvorm:	Inhoud R13
Ø	Ø	Ø	Ø		Ø
Ø	Ø	Ø	1		1
Ø	Ø	1	Ø		2
Ø	Ø	1	1		3
Ø	1	Ø	Ø		4
Ø	1	Ø	1		5
Ø	1	1	Ø		6
Ø	1	1	1		7
1	Ø	Ø	Ø		8
1	Ø	Ø	1		9
1	Ø	1	Ø		1Ø
1	Ø	1	1		11
1	1	Ø	Ø		12
1	1	Ø	1		13
1	1	1	Ø		14
1	1	1	1		15

e

Dit zijn dus nog lang geen echte ADSR karakteristieken zoals eerder beschreven, echter met bovenstaande envelope vormen zijn de meeste effecten al na te bootsen. Het is overigens wel mogelijk ASR omhullende samen te stellen met behulp van bovenstaande vormen. Dit wordt onder andere in de Forth applicatie (zie software beschrijving) toegepast.

De registers R14 en R15 zijn de registers waarin data staan, welke bestemd zijn voor respectievelijk poort A en B, of data welke op de poort werden ingelezen.

Hoe nu uiteindelijk een klank wordt opgewekt met behulp van al de beschikbare registers, zullen we bekijken bij de software-beschrijving. De kwaliteit van de geproduceerde muziek of geluidseffecten hangen namelijk geheel af van de kwaliteit van de software.

Alvorens over te gaan tot deze software-beschrijving nog een tweetal tabellen, welke een oversicht geven van respectievelijk de waarden welke in een frequentieregister (RØ/R1/R2/R3/R4/R5) moeten komen om een bepaalde toonhoogte te krijgen en de waarden welke in de registers R11 en R12 moeten om een bepaalde envelope-timing te krijgen.

Tabel 1 PITCH TABLE

Inhoud van de registers RØ-R1, R2-R3 en R4-R5.

noot \ oktaaf	Ø		1		2		3		4		5	
	R1	RØ	R1	RØ	R1	RØ	R1	RØ	R1	RØ	R1	RØ
A	4	116	2	58	1	29	Ø	142	Ø	71	Ø	36
A#	4	56	2	24	1	13	Ø	134	Ø	67	Ø	34
B	3	243	1	249	Ø	253	Ø	127	Ø	63	Ø	32
C	3	182	1	222	Ø	239	Ø	119	Ø	60	Ø	3Ø
C#	3	141	1	195	Ø	226	Ø	113	Ø	56	Ø	28
D	3	8Ø	1	17Ø	Ø	213	Ø	1Ø6	Ø	53	Ø	27
D#	3	36	1	146	Ø	2Ø1	Ø	1ØØ	Ø	5Ø	Ø	25
E	2	243	1	124	Ø	189	Ø	95	Ø	47	Ø	24
F	2	2Ø8	1	1Ø2	Ø	179	Ø	9Ø	Ø	45	Ø	22
F#	2	162	1	83	Ø	169	Ø	84	Ø	42	Ø	21
G	2	128	1	64	Ø	159	Ø	8Ø	Ø	4Ø	Ø	2Ø
G#	2	91	1	45	Ø	151	Ø	75	Ø	38	Ø	19

(In de tabel staat RØ en R1, deze kunnen dus ook zijn R2 en R3 of R4 en R5).

Tabel 2 ENVELOPE TIMING TABLE

Inhoud van de registers R11 en R12

t (seconden)	n	R11	R12
1/32	125	125	Ø
1/16	25Ø	25Ø	Ø
3/16	75Ø	24Ø	2
5/16	125Ø	23Ø	4
1/8	5ØØ	245	1
3/8	15ØØ	225	5
5/8	25ØØ	205	9
1/4	1ØØØ	235	3
3/4	3ØØØ	195	11
1/2	2ØØØ	215	7
1	4ØØØ	175	15
2	8ØØØ	95	31
4	16ØØØ	19Ø	62

Voor de overige timings geldt:

1 eenheid in R11/R12 komt overeen met Ø,ØØØ25 s. In de tabel is "n" het aantal eenheden.

Deze software beschrijving bestaat, zoals gezegd, uit twee delen, namelijk een aantal korte MC routines en een Forth applicatie.
Feitelijk is al deze software opgebouwd rond slechts twee zeer korte MC routines, namelijk één om data in de registers weg te schrijven en één om data uit de registers terug te lezen. Deze routines gebruiken een registernummer en een databyte of alleen een registernummer.
M.b.v. deze routines vult men dus register voor register, of leest men register voor register uit.
De routines doen het volgende:

(data naar register)
- zet BC1, BC2 en BDIR zodanig dat de bus gereed is om een registernummer te ontvangen (adres)
- zet het registernummer op DAØ-DA7
- zet BC1, BC2 en BDIR zodanig, dat de bus gereed is data te ontvangen
- zet de data op DAØ-DA7

(data uit register)
- zet BC1, BC2 en BDIR zodanig, dat de bus gereed is om een registernummer te ontvangen (adres)
- zet het registernummer op DAØ-DA7
- zet BC1, BC2 en BDIR zodanig, dat de bus gereed is om data te transporteren
- zet de Atom Via poort als input
- lees de Atom Via poort en schrijf in geheugen-lokatie
- zet de Atom Via poort als output

De functies van BDIR en BC1 zijn:

BDIR	BC1	BC2	
Ø	Ø	1	inactief
Ø	1	1	lees de PSG
1	Ø	1	schrijf naar PSG

(BC2 is dus altijd hoog).

2.1 MC routines.

De machinetaal-routines waar de registers mee gevuld en gelezen kunnen worden zijn de volgende:

```
: LLØ   LDA @ 112    zet 112 (binair Ø111ØØØØ) op
        STA # B8ØØ   poort B (van de VIA), dus zet BC1,BC2 en BDIR
        STX # B8Ø1   zet de inhoud van X op poort A (X bevat registernr.)
        LDA @ 96     zet 96 (binair Ø11ØØØØØ) op poort B
        STA # B8ØØ   dus zet BC1, BC2 en BDIR
        STY # B8Ø1   zet de inhoud van Y op poort A (data voor register)
        LDA @ 64     zet 64 (binair Ø1ØØØØØØ) op poort B
        STA # B8ØØ   (poort standby)
        RTS
: LL1   LDA @ 112    zet 112 op poort B
        STA # B8ØØ   dus prepareer BC1,BC2 en BDIR voor register
        STX # B8Ø1   zet de waarde van X op poort A (X bevat reg.nr.)
        LDA @ 8Ø     zet 8Ø op poort B (binair Ø1Ø1ØØØØ), zet BC1,BC2
        STA # B8ØØ   en BDIR voor lezen van PSG
        LDA @ ØØ     zet Ø in #B8Ø3, d.w.z. poort A wordt input
        STA # B8Ø3
        LDA # B8Ø1   zet de waarde van poort A (reg. inhoud) in Accumul.
```

```
STA #9Ø        zet accumulator in locatie #9Ø
LDA @64        zet 64 (binair Ø1ØØØØØØ) op poort B
STA #B8ØØ      (standby)
LDA @#FF       programmeer poort A weer als output poort
STA #B8Ø3
RTS
```

Routine LLØ verwacht een registernummer in X en de weg te schrijven data in Y. Routine LL1 verwacht een registernummer in X en leest de data uit dat register in lokatie #9Ø.
Om klanken te vormen, kunnen we met deze routines al uit de voeten.
Als je het al wilt proberen, voer dan bovenstaande routines in en assembleer ze, probeer dan eens het volgende BASIC programma:

```
1Ø   REM tone A
2Ø   ?#B8O3=#FF;?#B8Ø3=#FØ             zet poortA en (1/2)poortB als output
3Ø   F. X=Ø TO 15; Y=Ø;LINK LLØ;N.     zet alle registers Ø  (reset)
4Ø   X=1;Y=Ø;LINKLLØ                   vul register 1 met Ø
5Ø   X=Ø;Y=2ØØ;LINKLLØ                 vul register Ø met 2ØØ
6Ø   X=7;Y=254;LINKLLØ                 enable oscillator A op mixer A
7Ø   X=8;Y=15;LINKLLØ                  zet amplitude v. kanaal A op 15
8Ø   END
```

Is de werking duidelijk?, probeer ook eens andere toonhoogten en amplituden, probeer ook kanaal B en C aan te spreken. Experimenteer eens wat met ruis.

Deze routines zijn, zoals gezegd, de kern van alle software voor onze PSG. Bovenstaande methode is nog wat omslachtig, omdat je register voor register moet veranderen en invullen.
Hieronder volgt een listing van een machinetaal-routine waarmee alle PSG-functies bediend kunnen worden door simpelweg een link naar een bepaald labelnummer te geven. Achter de listing volgt een tabel waarin de functies van de verschillende onderdelen van de MC-routine staan beschreven.
Het gebruik is als volgt:

- de toonhoogten voor de oscillators plaatst men in respectievelijk A, B en C (dat zijn de BASIC variabelen !)
- de grondfrequentie van de ruisgenerator komt in variabele N
- de amplitude voor de kanalen A,B en C komen in de variabelen X,Y en Z
- de waarde voor de envelope timing komt in E
- na alle BASIC variabelen te hebben gevuld/aangepast kan men de gewenste routines "linken"

```
1ØØ   REM PSG functions
11Ø   P.$12;IN."Assembleren vanaf: "Q
12Ø   DIM LL36;F.I=Ø TO 36; LLI=Q ; N.
13Ø   P.$21;F.I=Ø TO 1 ; P=Q
14Ø   [ ; :LLØ LDA @ 112; STA #B8ØØ; STX#B8Ø1; LDA @96; STA#B8ØØ
15Ø      STY #B8Ø1; LDA @ 64; STA #B8ØØ; RTS
16Ø     :LL1 LDA @ 112: STA #B8ØØ; STX#B8Ø1; LDA @8Ø; STA#B8ØØ
17Ø      LDA @Ø; STA#B8Ø3; LDA#B8Ø1;STA#9Ø; LDA @64; STA#B8ØØ
18Ø      LDA @#FF; STA#B8Ø3;RTS
19Ø     :LL2 LDA@#FF; STA#B8Ø3; LDA @#FØ; STA#B8Ø2; LDA @#FF; STA#9Ø;LDX@16
2ØØ     :LL3 DEX; LDY @Ø;JSRLLØ; CPX @Ø; BNELL3;RTS
```

```
210        :LL4 JSR LL34; SBC @1; JSRLL35; RTS
220        :LL5 JSR LL34; SBC @2; JSRLL35; RTS
230        :LL6 JSR LL34; SBC@4 ; JSRLL35; RTS
240        :LL7 JSR LL34; SBC @8; JSRLL35; RTS
250        :LL8 JSR LL34; SBC @16; JSRLL35; RTS
260        :LL9 JSR LL34; SBC @32; JSRLL35; RTS
270        :LL10 JSR LL36; ADC @1; JSRLL35; RTS
280        :LL11 JSR LL36; ADC @2; JSRLL35; RTS
290        :LL12 JSR LL36; ADC @4; JSRLL35; RTS
300        :LL13 JSR LL36; ADC @8; JSRLL35; RTS
310        :LL14 JSR LL36; ADC @16; JSR LL35; RTS
320        :LL15 JSR LL36; ADC @32; JSR LL35; RTS
330        :LL16 LDX @6; LDY #32F; JSR LL0; RTS
340        :LL17 LDX @0; LDY #322; JSR LL0; LDX @1; LDY #33D; JSR LL0; RTS
350        :LL18 LDX @2; LDY #323; JSR LL0; LDX @3; LDY #33E; JSR LL0; RTS
360        :LL19 LDX @4; LDY #324; JSR LL0; LDX @5; LDY #33F; JSR LL0; RTS
370        :LL20 LDX @8; LDY #339; JSR LL0; RTS
380        :LL21 LDX @9; LDY #33A; JSR LL0; RTS
390        :LL22 LDX @10; LDY #33B; JSR LL0; RTS
400        :LL23 LDX @11; LDY #326; JSR LL0; LDX @12; LDY #341; JSR LL0; RTS
410        :LL24 LDX @0
420        :LL25 JSR LL1; LDA #90; JSR #F802; JSR #F7FD; JSR # F7FD
430         CPX @16; BEQLL26; INX; JMP LL25
440        :LL26 RTS
450        :LL27 LDX @13; LDY @4; JSRLL0; RTS
460        :LL28 LDX @13; LDY @1; JSRLL0; RTS
470        :LL29 LDX @13; LDY @13; JSRLL0; RTS
480        :LL30 LDX @13; LDY @11; JSRLL0; RTS
490        :LL31 LDX @13; LDY@10 ; JSRLL0; RTS
500        :LL32 LDX @13; LDY @12; JSRLL0; RTS
510        :LL33 LDX @13; LDY @8; JSRLL0; RTS
520        :LL34 SEC; LDX@7; LDA#90; RTS
530        :LL35 STA#90; TAY; JSRLL0;RTS
540        :LL36 CLC; LDX @7; LDA#90;RTS
550    ]
560    N.; P.$6; @=0; P.$14
570    F.I=0 TO 36
580    P."label: LL"I"   "LLI'
590    N.; P.$15
600    END
```

Tabel: Labels/PSG functies

Labelno.	Functie
LL2	reset alle registers (vullen met 0)
LL4	enable oscillator A op mixer A
LL5	enable oscillator B op mixer B
LL6	enable oscillator C op mixer C
LL7	enable ruis op mixer A
LL8	enable ruis op mixer B
LL9	enable ruis op mixer C
LL10	disable (uitschakelen) oscillator A op mixer A
LL11	disable oscillator B op mixer B
LL12	disable oscillator C op mixer C
LL13	disable ruis op mixer A
LL14	disable ruis op mixer B

LL15	disable ruis op mixer C
LL16	verander ruisfrequentie
LL17	verander oscillator frequentie A
LL18	verander oscillator frequentie B
LL19	verander oscillator frequentie C
LL20	verander amplitude kanaal A
LL21	verander amplitude kanaal B
LL22	verander amplitude kanaal C
LL23	verander envelope timing waarde
LL24	dump alle registerinhouden (RØ tot R15)
LL27	envelope golfvorm: eenmalige ATTACK
LL28	envelope golfvorm: eenmalige RELEASE
LL29	envelope golfvorm: ATTACK-SUSTAIN (aanhoudend)
LL3Ø	envelope golfvorm: RELEASE-SUSTAIN (aanhoudend)
LL31	envelope golfvorm: repeterende ATTACK RELEASE
LL32	envelope golfvorm: repeterende ATTACK
LL33	envelope golfvorm: repeterende RELEASE

De functie RESET (label LL2) moet altijd voorafgaan aan het verder vullen van de registers (ook de poorten A en B worden hierdoor geïnitieerd). Enkele voorbeelden: (assembleer natuurlijk eerst de routines hierboven)

```
100  REM modulation
110  LINK LL2
120  A=150;LINKLL17; B=151;LINKLL18
130  LINKLL4;LINKLL5
140  X=10;Y=10;LINKLL20;LINKLL21
150  F.I=1 TO 1000;N.
160  F.A=0 TO 1000;B=A;LINK LL17;LINK LL18;N.
170  LINK LL2; END

100  REM random noises
110  LINK LL2
120  LINKLL7;X=16;LINKLL20;E=300;LINKLL23
130  DO N=A.R.% 31; LINKLL16;LINKLL28; F. I=1 TO 100;N.
140  UNTIL 0

100  REM chords
110  LINK LL2
120  LINKLL4;LINKLL5;LINKLL6
130  X=16;Y=16;Z=16;LINKLL20;LINKLL21;LINKLL22
140  A=400;B=300;C=200;E=800;LINKLL23;LINKLL31
150  DO A=A-1;B=B-1;C=C-1;LINKLL17;LINKLL18;LINKLL19
160  UNTIL A=225
170  LINK LL2;END
```

Alvorens over te gaan tot de bespreking van de FORTH applicatie wil ik graag even opmerken dat de machinetaalroutines voor het schrijven en lezen van registerdata aangepast moeten worden indien men de aansluitingen voor BC1,BC2 en BDIR anders heeft gekozen. Mocht het geheel niet werken, ga dit dan in ieder geval even na. In de eerste twee MC-routines staat aangegeven welke binaire getallen op poort B worden gezet, pas deze waarden dus aan of koppel BC1, BC2 en BDIR zoals in de hardwarebeschrijving is geadviseerd.

(wordt vervolgd.)

NA DE BUFFERS (2 X 74LS244 EN 1 X 74LS245) GAAN DE ADRES- EN DATALIJNEN NAAR DE GEHEUGEN IC'S EN DE DECODERS (74LS138). PIN 1 VAN DE '245 KOMT VAN DE 2 X GEINVERTEERDE R/W LIJN (PIN 30 CONN) VIA PIN 13, 12, 3, 4 VAN IC2.
EN (PIN 19) VAN DE '245 KOMT VAN DE UITGANG (PIN 8) VAN EEN 3-INPUT NAND (IC1) WAARVAN PIN 9 MET +5 IS VERBONDEN, PIN 10 AAN DE KLOKPULS (02 PIN 29 CONN) EN PIN 11 MET EEN TWEEDE 3-INPUT NAND (OOK IC1 PIN 12) IS VERBONDEN.
OP PIN 1, 13, 2 (IC1) STAAN DE ADRESGEBIEDEN WAARIN DE '245 ENABLE DIENT TE ZIJN, RESP. AXXX, EXXX EN BFFF. DE EERSTE TWEE KOMEN UIT DE ADRESDECODER (IC7), DE LAATSTE UIT DE COMBINATIE BXXX EN XFFF (SAMENGEVOEGD IN IC6 PIN 12, 13, 11). XFFF ONTSTAAT IN IC9 WAARBIJ ALLE ADRESLIJNEN A0 T/M A11 HOOG DIENEN TE ZIJN. TEVENS WORDT OP PIN 15 (IC9) HET GEINVERTEERDE RW SIGNAAL AANGEBODEN.

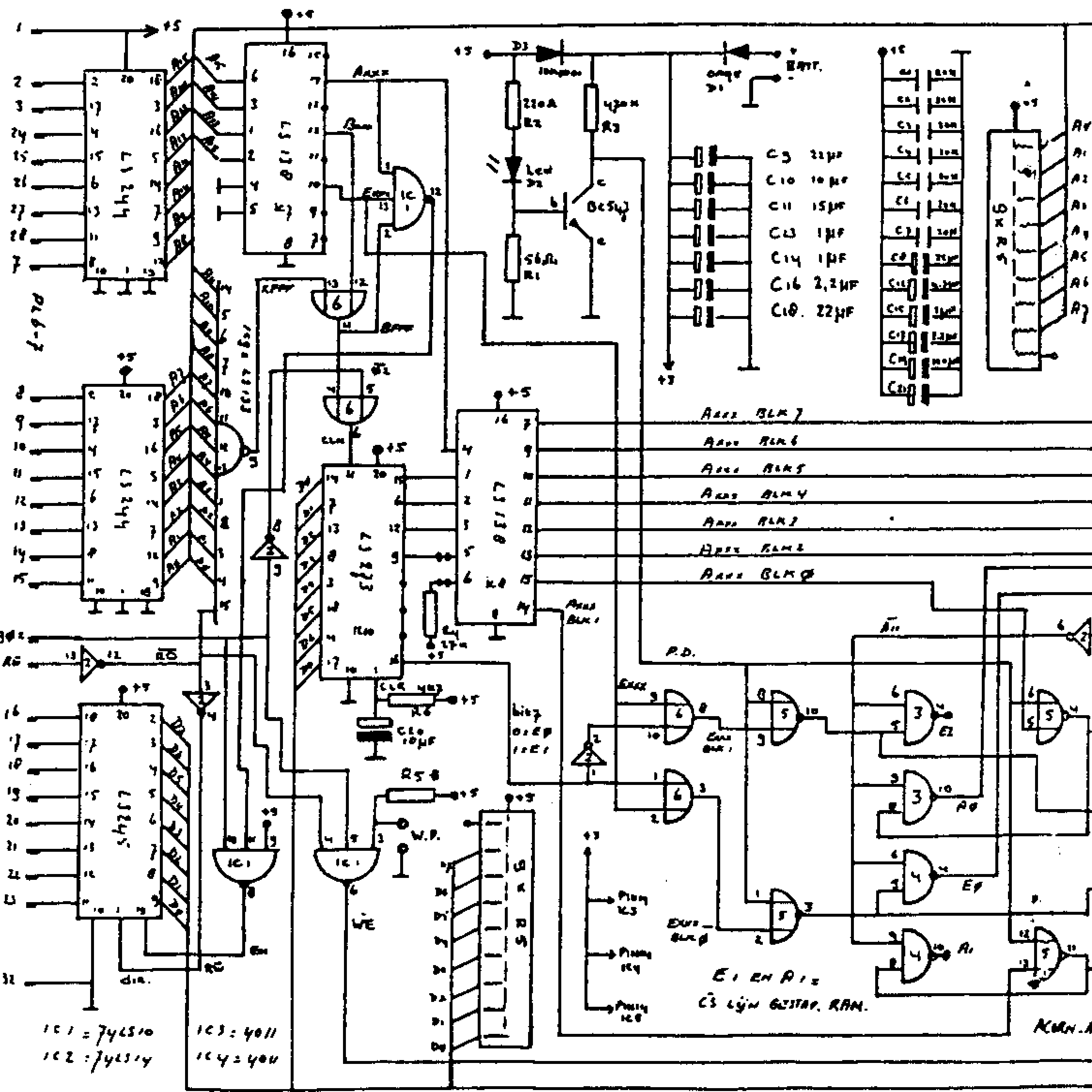
BFFF LEVERT TEVENS DE LATCHPULS VOOR IC10 PIN 11. DIT IN COMBINATIE MET EEN GEINVERTEERDE KLOKPULS 02 (VIA IC2 PIN 9, 8 EN IC6 PIN 4, 5, 6).
KOMT DEZE LATCHPULS (OP BFFF) DAN WORDT DE DATA VAN DE DATALIJNEN OP DE UITGANGEN GEZET EN VASTGEHOUDEN.
DE LATCHPULS EN DE CLR INGANG (IC10 PIN 1) ZIJN ACTIEF LAAG.
DE DECODER IC8 ('138) ZORGT ERVOOR DAT HET GEWENSTE BLOK OP AXXX WORDT INGESCHAKELD (BLOK 0 T/M 7).
BIT 7 (IC10 PIN 16) IS ERVOOR OM TE SCHAKELEN TUSSEN DE BEIDE BLOKKEN OP EXXX (E0 EN E1).
HET SIGNAAL WE UIT IC1 PIN 6 WORDT OPGEBOUWD UIT DE KLOKPULS 02 OP PIN 5 EN DE GEINVERTEERDE R/W OP PIN 4. PIN 3 ZIT VIA R5 (*) AAN DE +5. DOOR DEZE (PIN 3) MET DE MIN TE VERBINDEN WORDT 'WRITE PROTECT' INGESCHAKELD OP ALLE RAM'S.
U VINDT DEZE PUNTEN ONDER DE OPDRUK 'C8' OP DE PRINT.

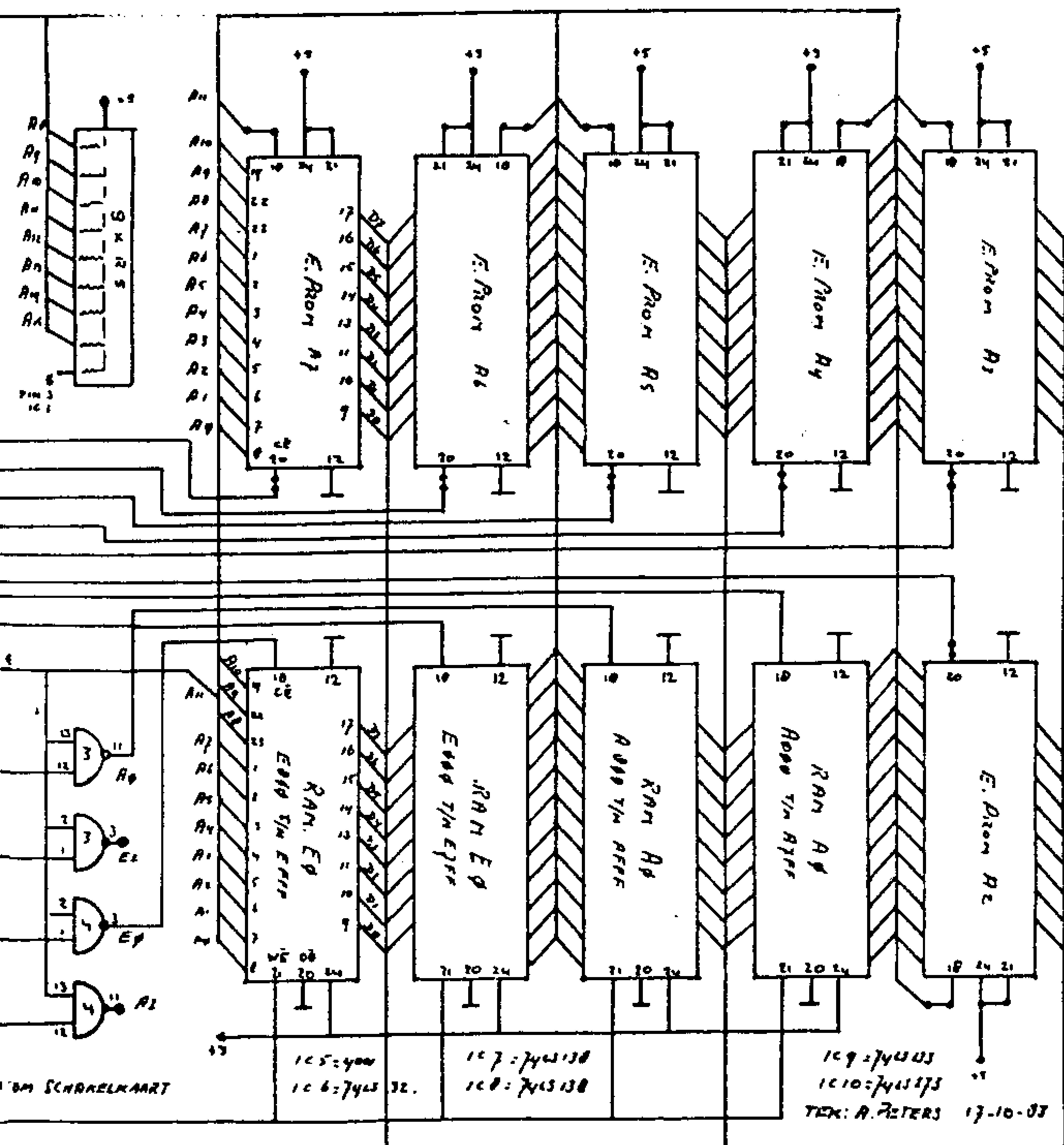
DE SCHAKELING MET DE TRANSISTOR IS DE 'POWER DOWN' SCHAKELING. IC5 IS DE EIGENLIJKE SCHAKELAAR NAAR 'POWER DOWN'.
BIJ INGESCHAKELDE VOEDING IS DE LIJN P.D. LAAG.
A11 EN A11 INVERS ZORGEN VOOR SPLITSING VAN DE 4K BLOKKEN T.B.V. DE 2K RAM'S.
R6 EN C20 (IC10 PIN 1) ZORGEN VOOR RESET BIJ INSCHAKELEN VAN DE VOEDING.
ALLE CMOS RAM'S ZIJN IN BATTERY BACKUP GESCHAKELD.
DE AANSLUITPUNTEN E1 EN A1 OP DE PRINT ZIJN DE CS LIJNEN VOOR DE GESTAPELDE RAM'S. E1 VOOR EXXX EN A1 VOOR AXXX.

IN DE BIJ DE PRINT GELEVERDE DOCUMENTATIE STAAN ENKELE FOUTEN:
. IC1 PIN 3, 4 EN 5 VERKEERD
. EXXX BLOK I EN EXXX BLOK II VERWISSELD; MOET ZIJN BLOK 1 EN BLOK 0
. TEKST "SCHAKELBIT7 LATCH PIN 16" HOORT BIJ ELKAAR (IC6 PIN 3), DUS NIET AFKOMSTIG VAN IC6 PIN 6
. BIJ IC5 PIN 5 EN PIN 13 STAAT RESP "CS A6 IC10" EN "CS A7 IC10"
  DIT MOET ZIJN RESP "AXXX BLK 0" EN "AXXX BLK 1"

BESCHRIJVINGEN VAN DE GEBRUIKTE IC'S:

IC1 = 3-INP NAND X 3	ALLE ING HOOG --> UITG LAAG
IC2 = HEX SMITT TRIGGER X 6	ING HOOG --> UITG LAAG
IC3 = IC4 = 2-INP NAND X 4	BEIDE ING HOOG --> UITG LAAG
IC5 = 2-INP NOR X 4	1 OF BEIDE ING HOOG --> UITG LAAG
IC6 = 2-INP OR X 4	1 OF BEIDE ING HOOG --> UITG HOOG
IC7 = IC8 = ADRESDECODER	UITG = ACTIEF LAAG
IC9 = 13-INP NAND	ALLE ING HOOG --> UITG LAAG
IC10= D-FLIPFLOP X 8	UITGANG=INGANG BIJ CLR HOOG EN CLK LAAG ALLE UITGANGEN 0 BIJ CLR (PIN1) LAAG


E1 EN A1 =
CS LIJN GESTAP. RAM.
IC1 = 74LS10
IC2 = 74LS14
IC3 = 4011
IC4 = 4011
C9 22μF
C10 10μF
C11 15μF
C13 1μF
C14 1μF
C16 2,2μF
C18 22μF
AXXX BLK 7
AXXX BLK 6
AXXX BLK 5
AXXX BLK 4
AXXX BLK 3
AXXX BLK 2
AXXX BLK Ø
BC547
BATT.
W.P.
WE
P.D.


E. Prom A7
E. Prom A6
E. Prom A5
E. Prom A4
E. Prom A3
RAM E0 E000 t/m E7FF
RAM E8 E800 t/m EFFF
RAM A0 A000 t/m A7FF
RAM A8 A800 t/m AFFF
E. Prom A2
IC 6: 74LS 32.
IC 7: 74LS138
IC 8: 74LS138
IC 9: 74LS 03
IC 10: 74LS173
TEK: A. PETERS 13-10-83

AAN DE HAND VAN DE WERKING VAN DE IC'S KUNT U ZELF NAGAAN HOE DE SCHAKELING VOOR DE CS LIJNEN VAN DE RAM'S WERKT. IK ZAL HIER VOLSTAAN MET EEN VOORBEELD:
STEL DAT BIT 7 LAAG IS (IC10 PIN 16) EN EXXX IS LAAG (IC7 PIN 10).
VAN IC6 IS DUS PIN 1 EN 2 LAAG. VOLGENS DE BESCHRIJVING IS DUS OOK PIN 3 LAAG.
VAN IC5 IS DUS PIN 1 EN 2 LAAG (P.D. LAAG). PIN 3 VAN IC5 IS DUS HOOG.
STEL NU OOK A11 LAAG, DAN IS DUS A11 INVERS HOOG. DIT IS PIN 6 VAN IC4.
OOK PIN 5 VAN IC4 WAS HOOG WANT DEZE LIGT AAN PIN 3 VAN IC5. DIT BETEKENT WEER DAT PIN 4 VAN IC4 LAAG IS. DIT IS DE CS LIJN VAN RAM E0 (E000-E7FF).
U ZIET DAT UITLEG VAN ALLE CS LIJNEN EEN BOEKJE APART ZOU VORMEN.
HOPELIJK VERSCHAFFEN SCHEMA EN TOELICHTING U VOLDOENDE INZICHT IN DE WERKING.

SOFT WRITE PROTECT OP DE SCHAKELKAART

A.PETERS

ONDER OPDRUK C8 BEVINDEN ZICH 2 EILANDJES, WELKE VERBONDEN DIENEN TE WORDEN OM ALLE RAM WRITE PROTECT TE KUNNEN SCHAKELEN. DIT KAN MET EEN DRAADBRUG OF MET EEN SCHAKELAAR. DIT IS LASTIG ALS DE PRINT ZICH IN DE KAST BEVINDT.

AANGEZIEN ER OP IC10 (74LS273) VAN DE SCHAKELKAART VOLDOENDE UITGANGEN VRIJ ZIJN, KAN DIT OOK SOFTWAREMATIG GEBEUREN. VERWIJDER SCHAKELAAR OF DRAADBRUG (INDIEN AANWEZIG). VERBIND IC1 PIN3 (74LS10) MET IC10 PIN5. DAT IS ALLES!!! HET SCHEMA MET TOELICHTING KAN E.E.A. VERDUIDELIJKEN, DEZE INFORMATIE IS IN DE REGIO ARNHEM AANWEZIG. BIJ INSCHAKELEN IS DE SITUATIE "WRITE PROTECT".

STURING GEBEURT DOOR HET VULLEN VAN HEX BFFF MET DE VOLGENDE WAARDEN:

WRITE-PROTECT		NO-WRITE-PROTECT	
0=RAM A0	128=RAM A0	64=RAM A0	192=RAM A0
1=RAM A1	129=RAM A1	65=RAM A1	193=RAM A1
2=EPR A2	130=EPR A2	66=EPR A2	194=EPR A2
3=EPR A3	131=EPR A3	67=EPR A3	195=EPR A3
4=EPR A4	132=EPR A4	68=EPR A4	196=EPR A4
5=EPR A5	133=EPR A5	69=EPR A5	197=EPR A5
6=EPR A6	134=EPR A6	70=EPR A6	198=EPR A6
7=EPR A7	135=EPR A7	71=EPR A7	199=EPR A7
+BLOK E0	+BLOK E1	+ BLOK E0	+BLOK E1

ELEKTUUR komt in JUNI of STEPTEMBER uit met een direkt gekoppelde telefoon-modem. Misschien is het niet onverstandig deze ontwikkeling af te wachten. De kosten van dit modem liggen tussen f. 300,00 en f. 400,00.

EDIT EN DEBUG ZIJN MACHINETAAL PROGRAMMA'S OM DE PROGRAMMEUR TE HELPEN EFFICIENTE EN GESTRUCTUREERDE PROGRAMMA'S TE SCHRIJVEN.
EDIT IS EEN GEAVANCEERDE TEXT EDITOR : DEBUG IS EEN HULPMIDDEL VOOR HET DEBUGGEN VAN MACHINE TAAL PROGRAMMA'S.
SAMEN ZITTEN ZE IN EEN 2532 EPROM DIE OP DE PLAATS HOORT VAN IC-24 IN DE ATOM.EDIT EN DEBUG ZITTEN DUS IN HET GEHEUGEN VAN AF #A000 TOT EN MET #AFFF.
ALS DE FLOATING POINT (EP)ROM AANWEZIG IS DAN IS EDIT OP TE STARTEN MET 'EDIT' EN DEBUG OP TE STARTEN MET 'DEBUG'.ALS DE F.P. ROM NIET IN DE COMPUTER ZIT DAN IS EDIT OP TE STARTEN MET 'LINK #A009' EN DEBUG IS DAN OP TE STARTEN MET 'LINK #A013'.
12-K OF MEER GEHEUGEN IS WEL NOODZAKELIJK WIL DE TOOLBUG HET GOED DOEN.
(RET) : BETEKENT GEVOLGD DOOR RETURN
0<N<256 DEF 1 : BETEKENT DAT N TUSSEN 0 EN 256 MAG LIGGEN.ALS JE NIKS VOOR N INTIKT DAN 'DEFAULT' (WORDT) DEZE 1, IN DIT VOORBEELD DAN.

EDIT

DE HUIDIGE REGEL (ENG: CURRENT LINE) IS DE REGEL VAN EEN PROGRAMMA WAAR HELEBOEL EDIT COMMANDO'S VAN UIT GAAN.HUIDIGE REGEL WORDT AFGEKORT MET HR.
HET WOORD 'TEXT' IS HET ENGELS VOOR EEN (BASIC)PROGRAMMA IN HET GEHEUGEN.
NA DAT EDIT IS AAN GEZET , VERSCHIJNT DE VOLGENDE BOODSCHAP OP HET SCHERM :

```
EDIT  29-2902  82-8202
?
```

DE 29-2902 HOUDT IN DAT HET HOOFD GEHEUGEN BEGINT VANAF #2900 EN DAT TOP VAN DE TEXT IN HET HOOFD GEHEUGEN #2902 IS.DE 82-8202 HOUDT IN DAT HET BUFFER GEHEUGEN BEGINT VANAF #8200 EN DAT TOP VAN DE TEXT IN HET BUFFER GEHEUGEN #8202 IS.
DE ? IS DE PROMPT (ZO HEET DAT DING) VOOR EDIT , EN WIJST ER OP DAT WAT JE NU INTIKT GEINTERPRETEERD WORDT DOOR EDIT.
BIJNA ALLE EDIT COMMANDO'S ZIJN ENKELE LETTERS.DE MEESTE VAN DEZE COMMANDO'S ZIJN GERICHT OP DE HR (HUIDIG REGEL).ALS JE NIET ZEKER WEET WAT DE HR IS , TIK DAN L IN DAN VERSCHIJNT DE HR OP HET SCHERM.
ALS JE GEWOON OP RETURN RETURN DRUKT DAN WORDT DE LAATST INGETIKTE COMMANDO HERHAALD.
HET IS AAN TE RADEN OM DE VOLGENDE COMMANDO'S ONDER DE KNIE TE KRIJGEN VOOR DAT ECHT GAAT WERKEN MET EDIT:

```
L --> LIST
N --> NEXT
H --> HUNT
S --> SUBSTITUTE
A --> AUTO
+ --> ADD
D --> DELETE
Q --> QUIT
```

HIER VOLGT NU GEDETAILLEERDE LIJST VAN EDIT COMMANDO'S:
A
An AUTO 1<n<256 DEF 1
GENEREERT AUTOMATIES REGELNUMMERS, DOOR n BIJ HET HR REGELNUMMER OP TE TELLEN. AUTO STOPT ALS JE GEWOON OP RTEURN DRUKT OF ALS n + HR-REGELNR. = BESTAAND REGELNR. . DUS JE KAN NIET PERONGELUK EEN AL BESTAANDE REGEL UITWISSEN.
B BOTTOM
DE LAATSTE REGEL VAN DE TEXT WORDT NU DE HR.
B=hh SET START OF BUFFER MEMORY
STEL hh=#8C DAN BEGINT HET BUFFER GEHEUGEN NU VANAF #8C00. hh TIK JE IN ALS EEN HEX-GETAL MAAR ZONDER MATJE(#). MAAR KIJK UIT WANT EDIT KIJKT NIET NA OF JE EEN PROGRAMMA IN HET HOOFD GEHEUGEN OVERSCHRIJFT OMDAT JE HET BUFFER GEHEUGEN LAAT BEGINNEN IN HET HOOFD GEHEUGEN.
C/STR1/STR2/n CHANGE 0<n<256 DEF 1
VANAF DE HR TOT EN MET n REGELS VERDER WORDT STR1 VERVANGEN DOOR STR2.
BV: C/RETURN/R./25
NU WORDT IN DE VOLGENDE 25 REGELS RETURN VERVANGEN DOOR R.
Dn DELETE 0<n<256 DEF 1
HAALT DE REGELS VANAF DE HR TOT EN MET n REGELS VERDER WEG.
DM DELETE MARKED TEXT
VERWIJDERT DE TEXT VANAF REGEL M1 TOT EN MET REGEL M2. GEBRUIK LM OM TE ZIEN WAT DE GEMARKTE TEXT IS. ZIE OOK M1 EN M2.
D. DELETE ALL FOLOWING TEXT
HAALT DE TEXT WEG VAN AF DE HR. EDIT ZET DUS EEN #FF NEER NET NA DE HR.
D=c CHANGE DELIMITER
HET KARAKTER DAT JE GEWOONLIJK TUUSEN TWEE COMMANDO'S NEER ZET OM ZE TE SCHEIDEN (GEWOONLIJK DE ';') HEET EEN DELIMITER. DEZE DELIMITER IS VOOR DE EDIT GEWOONLIJK + MAAR MET DIT COMMANDO KAN JE DE EDIT DELIMITER VERANDEREN IN KARAKTER c.
BV: D=:
NU IS DE EDIT DELIMITER DUS :
En EXTRACT 0<n<256 DEF 1
HAALT DE TEXT VANAF DE HR TOT EN MET n REGELS VERDER WEG UIT HET HOOFD GEHEUGEN EN ZET DEZE NEER IN HET BUFFER GEHEUGEN.
H/STR/n HUNT 0<n<256 DEF 1
LIST DE VOLGENDE n REGELS VANAF DE HR DIE STR BEVATTEN.
In INSERT 0<n<256 DEF 1
HAAL DE TEXT UIT HET BUFFER GEHEUGEN EN ZET DEZE TUSSEN DE HR EN DE REGEL DAARNA. n IS DE STAP VAN DE REGELNR.S VAN DE TEXT UIT HET BUFFER GEHEUGEN. ZONODIG WORDT DE TEXT GERENUMBERD.
Ln LIST 0<n<256 DEF 1
LIST DE VOLGENDE n REGELS VANAF DE HR. HIERNA IS DE HR DE LAATST GELISTE REGEL.
L-n LIST 0<n<256 DEF 1
LIST DE VOORGAANDE n REGELS VOOR DE HR. HIERNA IS DE HR DE EERST GELISTE REGEL.
LA LIST AROUND
LIST DE 10 REGELS OM DE HR INCLUSIEF DE HR.

LB LIST BACKWARDS
HET ZEFDE ALS L-10
LF LIST FORWARDS
HET ZELFDE ALS L10
LM LIST MARKED TEXT
LIST DE TEXT VANAF REGEL M1 TOT EN MET REGEL M2.ZIE OOK M1 EN M2.
L. LIST
LIST DE HELE TEXT (HET HELE PROGRAMMA DUS).
L.n LIST LINE
LIST REGEL n.ALS REGEL n NIET BESTAAT DAN WORDT DE EERSTE REGEL NA n DE HR. BESTAAT REGEL n WEL DAN WORDT REGEL n DE HR.n IS DUS EEN REGELNR.
M1 MARK TEXT
M2
ALS JE M1 (OF M2) INTIKT DAN WORDT DE HR M1 (OF M2).M1 MOOET WEL KLEINER ZIJN DAN M2.VOOR GEBRUIK BY LM,DM EN EM.
Nn NEXT 0(n(256 DEF 1
DE n-DE REGEL NA DE HR WORDT DE NIEUWE HR.
N-n NEXT 0(n(256 DEF 1
DE n-DE REGEL VOOR DE HR WORDT DE NIEUWE HR.
NEW NEW
EEN #FF WORDT VOOR DE EERSTE REGEL VAN DE TEXT GEZET.
OLD OLD
HAALT DE EVENTUELE #FF VOOR DE EERSTE REGEL WEG.
ON ENABLE PRINTER
ZET PRINTER AAN
OFF DISABLE PRINTER
ZET PRINTER UIT
Q QUIT
TERUG NAAR ATOM BASIC
Rn RENUMBER 0(n(256 DEF 1
DE TEXT NA DE HR WORDT GERENUMBERD IN STAPPEN VAN n .HIERNA IS DE HR DE LAATST GERENUMBERDE REGEL.
Rn,m RENUMBER 0(n(256 DEF 1
EERST KRIJGT DE HR HET REGELNR m DAARNA WORDT DE TEXT GERENUMBERD IN STAPPEN VAN n.
Rn,m,F FULL RENUMBER
DE EERSTE REGEL KRIJGT REGELNR m EN DAN WORDT DE HELE TEXT GERENUMBERD.
S/STR1/STR2/n SUBSTITUTE 0(n(256 DEF 1
VERVANGT STR1 DOOR STR2 n MAAL IN DE HR.
T TOP
DE HR WORDT DE EERSTE REGEL VAN DE TEXT.
X EXCHANGE
HET HOOFD GEHEUGEN WORDT HET BUUFER GEHEUGEN EN HET BUFFER GEHEUGENX EXCHANGE
HET BUFFER GEHEUGEN WORDT HET HOOFD GEHEUGEN EN HET HOOFD GEHEUGEN WORDT HET BUFFER GEHEUGEN.
Z STORE COMMAND LINE
DE COMMANDO'S DIE ACHTER DE Z WORDEN IN GETIKT , WORDEN OPGESLAGEN.DEZE COMMANDO'S WORDEN UITGEVOERD ALS JE - INTIKT.
- INVOKE STORED COMMAND LINE
DIT COMMANDO VOERT DE DOOR Z OPGESLAGEN COMMANDO'S UIT.DE HR

WORDT EEN REGEL OPGESCHOVEN.
+STR ADD
STR WORDT BY DE HR OPGETELD.
* DOS COMMAND
EEN OPERATING SYSTEM COMMANDO (*SAVE OF *LOAD ENZ.).
<XYZ LOAD
LOAD"XYZ"
>XYZ SAVE
SAVE"XYZ"
? INFORMATION
HET BEGIN ADRES EN TOP VAN DE PROGRAMMAS IN HET HOOFD GEHEUGEN EN HET BUFFER GEHEUGEN WORDEN OP HET SCHERM GEZET.
ALS JE GEWOON OP RETURN DRUKT DAN WORDT DE LAATST INGETIKTE COMMANDO HERHAALT.
ALS JE IETS INTIKT DAT MET EEN GETAL BEGINT DAN NEEMT EDIT DAT AAN ALS EEN BASIC REGEL , EN ZET DEZE DAN OOK OP DE GOEIE PLAATS IN HET GEHEUGEN.

EDIT COMMANDO'S

An	AUTO
B	HR=LAATSTE REGEL
B=hh	BUFFER GEHEUGEN BEGINT VANAF #hh00
C/STR1/STR2/n	VERANDER STR1 IN STR2 IN DE VOLGENDE n REGELS
Dn	DELETE DE VOLGENDE n REGELS
DM	DELETE DE GEMARKTE TEXT
D.	DELETE DE TEXT VANAF DE HUIDIGE REGEL
D=c	VERANDER DE DELIMITER
En	ZET DE VOLGENDE n REGELS IN HET BUFFER GEHEUGEN
EM	ZET DE GEMARKTE TEXT IN HET BUFFER GEHEUGEN
H/STR/n	LIST DE VOLGENDE n REGELS DIE STR BEVATTEN
In	VOEG TEXT TUSSEN IN VANUIT HET BUFFER GEHEUGEN
Ln	LIST DE VOLGENDE n REGELS
L-n	LIST DE VORIGE n REGELS
LA	LIST 10 REGELS OM DE HR
LB	L-10
LF	L10
M1	MARKEER DE HR ALS M1
M2	MARKEER DE HR ALS M2
Nn	HR=HR+n REGELS
N-n	HR=HR-n REGELS
NEW	NEW
OLD	OLD
ON	PRINTER AAN
OFF	PRINTER UIT
Q	TERUG NAAR ATOM BASIC
Rn	RENUMBER IN STAPPEN VAN n
Rn,M	RENUMBER IN STAPPEN VAN n VANAF M
Rn,M,F	RENUMBER DE HELE TEXT IN STAPPEN VAN n VANAF M
S/STR1/STR2/n	VERVANG STR1 DOOR STR2 n MAAL IN DE HR
T	HR=EERSTE REGEL

Y BUFFER GEHEUGEN=HOOFD GEHEUGEN, HOOFD GEHEUGEN=BUFFER GEHEUGEN
Z SLA COMMANDO REGEL OP
- VOER COMMANDO REGEL UIT
+STR TEL STR OP BY DE HR
(XYZ LOAD"XYZ"
)XYZ SAVE"XYZ"
? INFORMATIE

DEBUG

ALS DEBUG WORDT AANGEZET DAN WORDT DE BRK VECTOR (BRKVEC IN #202 EN #203) VERANDERD , DEZE ZAL NU NAAR DE DEBUG BRK ROUTINE WIJZEN. DEZE ROUTINE SLAAT DE 6502 REGITERS OP EN BEELD DEZE AF OP HET SCHERM.
DE GEHEUGEN PLAATSEN DIE DE WAARDEN VAN D 6502 REGISTERS, ACCU, ENZ. BEVATTEN , WORDEN 'IMAGES' GENOEMD (ENG: IMAGE=BEELD).
IN PLAATS VAN DE GEWONE) PROMPT ZIE JE - ALS PROMPT.
HB=HUIDIGE BYTE ; HELEBOEL DEBUG COMMANDO'S ZIJN GERICHT OP DEZE HUIDIGE BYTE.
HIER VOLGT NU EEN GEDETAILLEERDE LIJST VAN DEBUG COMMANDO'S.

A ACCUMULATOR
X X REGISTER
Y Y REGISTER
P STATUS REGISTER
DEZE 4 COMMANDO'S LATEN DE ACCU EN DE REGISTER IMAGES ZIEN VAN DE 6502 ZOALS ZE WAREN SINDS DE LAATSTE BRK.
BV: A (RET)
DAN VERSCHIJNT ER BV:
3C ?
DIE ? VRAAGT WAT JE WILT DOEN. ALS JE OP RETURN DRUKT DAN GAAT DEBUG GEWOON VERDER. MAAR ALS JE EEN HEX GETAL INTIKT , ZONDER MATJE (#) , DAN WORDT DE BETREFFENDE REGISTER VERANDERD IN HET GETAL DAT JE INTIKTE.
BV: 3C ?FF
NU ZAL DE BETREFFENDE REGISTER #FF BEVATTEN.

R REGISTERS
DIT COMMANDO LAAT DE 6502 ACCU, REGISTER, PROGRAM COUNTER EN STACK POINTER IMAGES ZIEN ZOALS ZE WAREN NA DE LAATSTE BRK.
B BREAKPOINT
LAAT ZIEN OP WELK TIJDELIJK ADRES DE DEBUG BRK (#00) STAAT. NA DE ? KAN JE DIT ADRES VERANDEREN. DAN WORDT IN HET OUWE BRK ADRES TERUG GEZET WAT ER OORSPRONKELIJK STOND EN WAT ER IN HET NIEUWE BRK ADRES STAAT WORDT OPGESLAGEN. IN HET NIEUWE BRK ADRES KOMT NU DE DEBUG BRK TE STAAN.
Cn DISASSEMBLE 0<n<256 DEF15
DISASSEMBLEERD n 6502 INSTRUCTIES VAN AF DE HB. NA DIT COMMANDO IS DE HB DE EERSTE BYTE VAN HET LAATST GEDISASSEMBLEERDE INSTRUCTIE.

C,H DISASSEMBLE
DISASSEMBLEER VANAF DE HB TOT EN MET ADRES H.
BV: C,3C00 (GEEN # NODIG)
Dn DUMP MEMORY 0<n<256 DEF 7
BEELDT n PAREN REGELS AF. DE EERSTE REGEL BEVAT HET ADRES VAN DE EERSTE BYTE EN EEN AANTAL BYTES AFGEBEELD IN HEX. DE TWEEDE REGEL BESTAAT UIT DE ASCII KARAKTERS VAN DE BETREFFENDE BYTES. HET AANTAL BYTES PER REGEL WORDT BEPAALD DOOR FORMAT.
D,H DUMP
DUMP VANAF DE HB TOT EN MET ADRES H.
F FORMAT
HIERMEE KAN JE HET DUMP AANTAL BYTES PER REGEL GETAL VERANDEREN. BV: F (RET)
DAN VERSCHIJNT ER BV:
19 ?
NA DE ? KAN JE EIGEN FORMAT GETAL INTIKKEN.
G GO
DE 6502 REGISTERS EN ACCU WORDEN GELADEN MET HUN IMAGES EN DE 6502 SPRINGT NAAR DE HB OM DE ROUTINE DIE DAAR BEGINT UIT TE VOEREN. BY DE EERSTE BRK WORDEN DE REGISTERS ENZ. IN HUN IMAGES GEZET. DE HB IS NU HET ADRES WAAR DE BRK (TIJDELIJK) STOND.
M MEMORY
LAAT DE INHOUD EN HET ADRES ZIEN VANAF DE HB , TOT DAT JE OP RETURN DRUKT NA DE ? . ALS JE OP DE SPATIE BALK DRUKT NA DE ? DAN WORDT DE VOLGENDE BYTE EN ADRES AFGEBEELD. NA DE ? KAN JE NATUURLIJK OOK WEER EEN GETAL INTIKKEN OM DE INHOUD VAN DE BETREFFENDE BIJTE TE VERANDEREN.
ON PRINTER AAN
OFF PRINTER UIT
Q TERUG NAAR ATOM BASIC
W WHERE
LAAT ZIEN OP WELK ADRES DE HB ZIT. NA DE ? KAN JE DEZE VERANDEREN NAAR EEN ANDER ADRES.

DEBUG COMMANDO'S

A LAAT DE ACCU IMAGE ZIEN
X LAAT DE X REGISTER IMMAGE ZIEN
Y LAAT DE Y REGISTER IMAGE ZIEN
P LAAT DE STATUS REGISTER IMAGE ZIEN
B LAAT HET BRK ADRES ZIEN
C DIASSEMBLE
D LAAT DE INHOUD VAN HET GEHEUGEN ZIEN HEX EN ASCII
F DUMP BYTES PER REGEL GETAL
G LINK NAAR DE HB
M LAAT DE INHOUD VAN HET GEHEUGEN ZIEN
ON PRINTER AAN
OFF PRINTER UIT
Q TERUG NAAR ATOM BASIC
R LAAT DE ACCU REGISTER SP EN PC IMAGES ZIEN
W LAAT HET ADRES ZIEN VAN DE HB

ALS JE GEWOON OP RETURN DRUKT DAN WORDT DE LAATST INGETIKTE COMMANDO HERHAALT.

Het SOS programma is een uitgebreid operating-system voor de schakelkaart. In Acorn Nieuws van augustus 1983 is dit systeem reeds geintroduceerd. Helaas was het programma alleen met het cassette operating system te gebruiken. Dit komt omdat de schakelkaart in z'n oorspronkelijke vorm de geheugenblokken A000 en C000 schakelt en aangezien het Diskette operating system in het E000 blok verblijft lopen beide systemen elkaar voor de voeten. Daar in vele regio's het E000 blok wordt vervangen door het 1000 blok wordt in onderstaand verhaal hier ook vanuit gegaan. Het SOS programma is nu geheel relatief geschreven en kan dus overal heen geassembleerd worden. In deze uiteenzetting wordt dat #1800.

Aangezien voor vele leden de schakelkaart nog een mysterie is wordt elders de werking uitgelegd.(op pag 49).

Het SOS programma bevat de volgende drie hoofdonderdelen:
1. Een eigen reset-routine.
2. Een eigen zoek-routine met eigen statements.
3. Het schakel-systeem om de verschillende blokken te selekteren.

In de volgende bespreking herhalen we niets wat reeds besproken is in het augustus nummer. Het systeem op zich is ongewijzigd gebleven op een punt na. Dit is namelijk de softtool herkenning. Aangezien de softtool bij het niet herkennen van een statement met een direkte jump naar #F8EF springt oftewel de DOS interpretatie-routine wordt de DOS dus buiten werking gesteld. Aangezien dit niet de bedoeling is voor de Dossisten is deze routine eruit gehaald.

1.De reset routine.
Bij de reset routine van het SOS systeem wordt de computer om de tuin geleid en kijkt op #EFFC en #EFFD. Dit gebeurd dus door het gewijzigde bootstrapprintje. Hierop staat nu dus de DOS-ROM maar daar zit nog ruimte in om deze vectoren in op te bergen. In de orginele ATOM-versie is de ROM na #EF40 leeg. In andere versies in combinatie met een TEAC-drive zit er nog een software-matige bootstrap in. Maar deze is ook nog in te korten. Door de monitor-rom te wijzigen is het hele bootstrap printje overbodig maar daarover een andere keer meer.
Het reset adres voor het SOS systeem op #1800 is #1800. Hierop staat een direkte jump naar de eigenlijke reset-routine. Dit is gedaan om bij een wijziging van het SOS programma niet een DOS-rom weer te moeten herprogrammeren. Bij reset wordt automatisch de DOS en Charon ingeschakeld. Bij het assembleren wordt de vraag gesteld waar de functie-statements van Charon moeten worden opgeslagen. Dit kan bijv. #95 zijn. Tevens wordt bij elke reset op #17FD het testbyte #E3 gezet en op #17FE en #17FF het indirekte jump-adres naar de SOS zoekroutine.

2.De eigen zoekroutine.
De zoekroutine is aangepast voor het zoeken in het geheugen beneden #8000. Immers in de hogere zoek routine's wordt met een

BMI op het hoge byte startadres getest hetgeen nu niet meer kan. Daarom is in de statement tabel tussen het statement en het startadres een #00 geplaatst. De tabel ziet er dan als volgt uit:
(statement) #00 (startadres)
Indien een statement niet gevonden wordt, wordt automatisch naar het schakelsysteem gesprongen. De zoekroutine wordt aangeroepen vanuit de Floating-point en de gewijzigde routine ziet er dan als volgt uit:

```
D4AF: LDA #17FD   Laadt testbyte SOS.
D4B2: CMP @#E3    vergelijk.
D4B4: BEQ #D4C0   indien juist dan naar SOS.
D4B6: LDA #A001   laadt testbyte JOS-Box.
D4B9: CMP @#BF    vergelijk.
D4BB: BNE D440    indien onjuist terug naar interpreter.
D4BD: JMP #A002   ga naar de JOS-BOX.
D4C0: JMP(#17EF)  ga naar SOS zoek routine.
```

Aanpassing van de F.P. ROM wordt dan:

D4B0	was 00	wordt FD		D4BC	was 03	wordt 83
D4B1	A0	17		D4C0	4C	6C
D4B3	40	E3		D4C1	02	FE
D4B4	D0	F0		D4C2	A0	17

Beschrijving SOS statement staan in het augustus nr. Van Auto kan nog vermeld worden dat de regelnummering wordt ingeschakeld met CRT-A en uitgeschakeld met wederom een CTR-A of een ESC. De gebruikte zero-page adressen zijn gewijzigd in #8C t/m #8F.

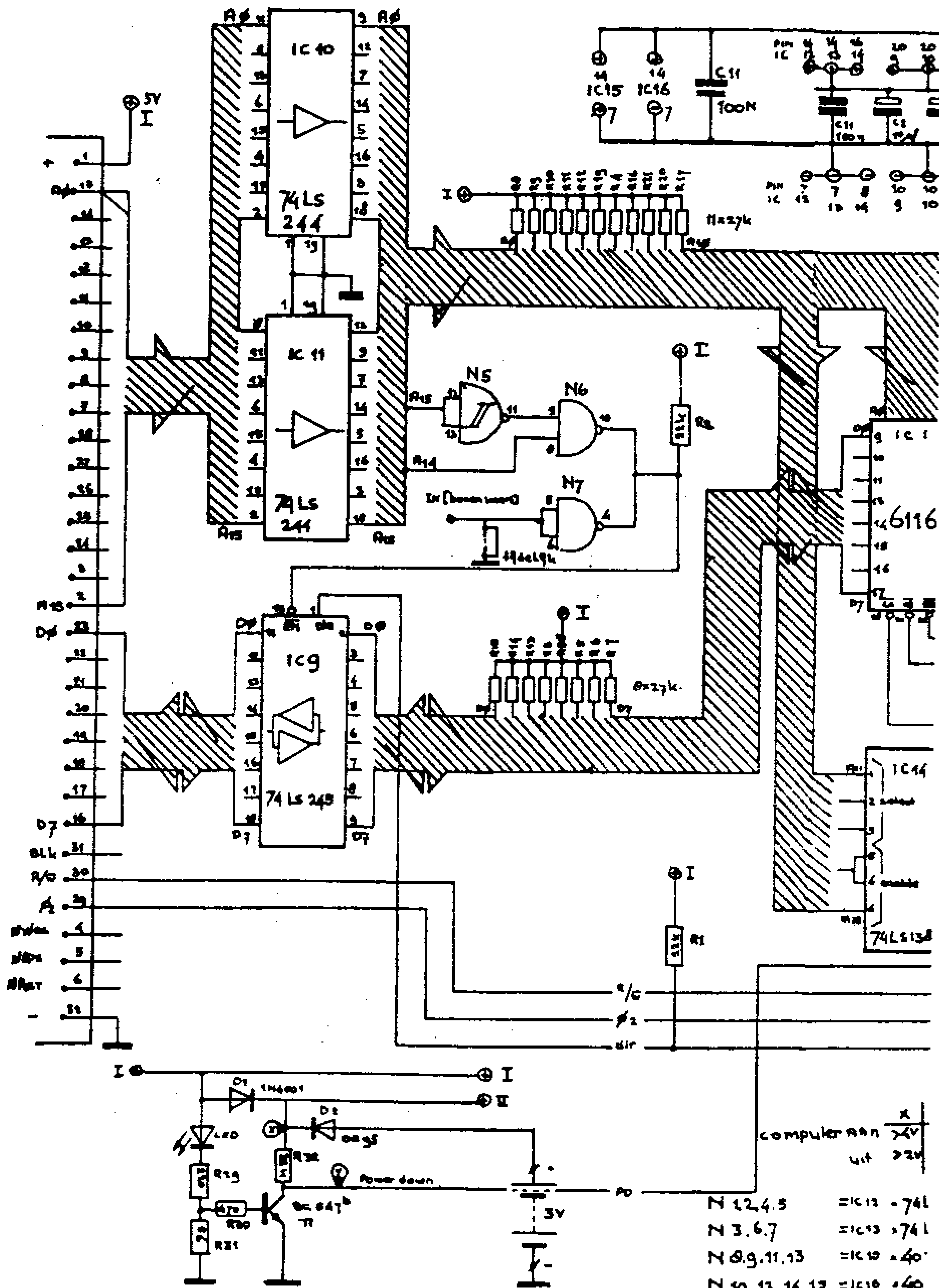
3. Het Schakelkaart Operating Sytem.

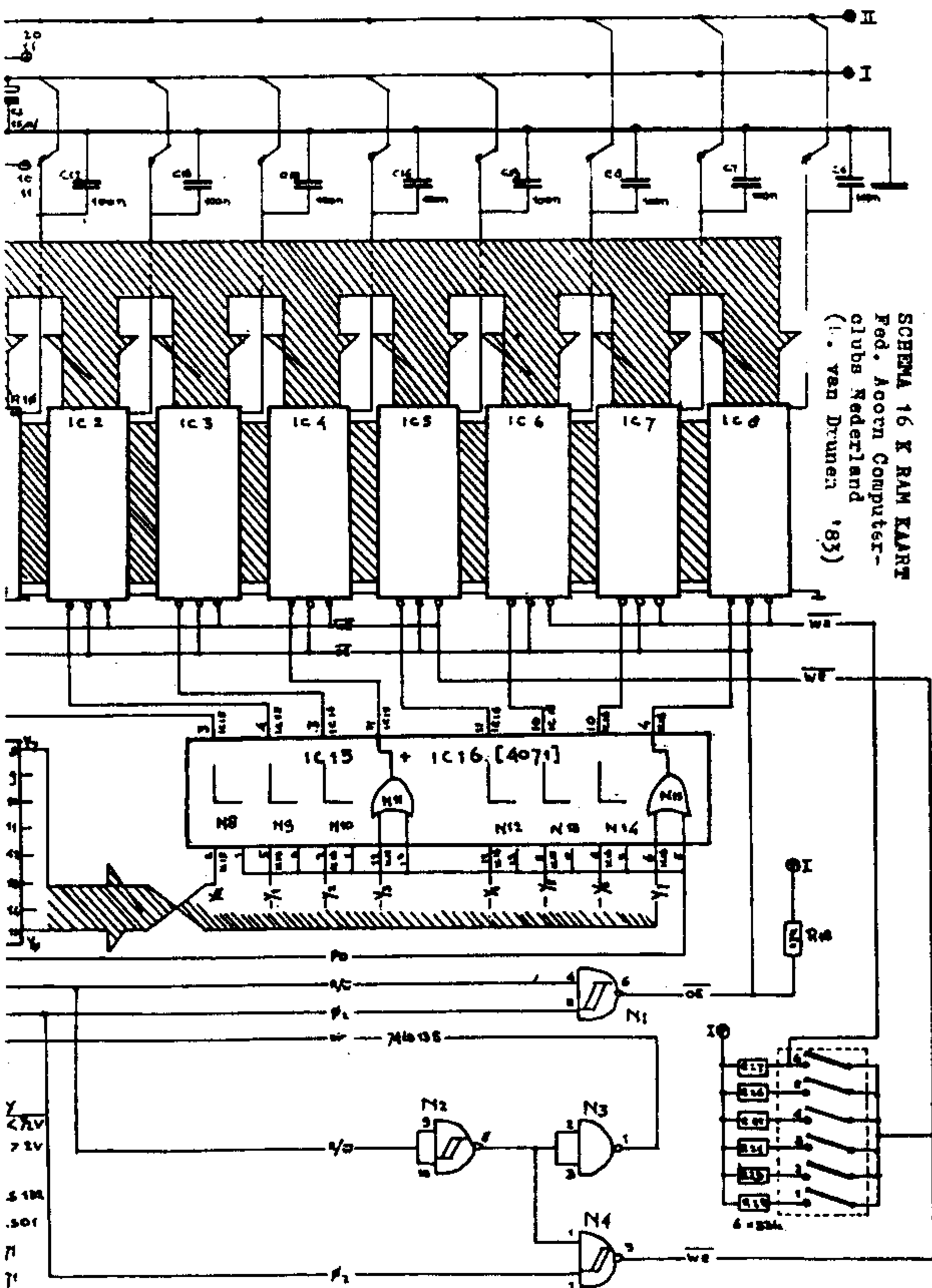
Indien het gezochte statement ook niet in de SOS statement-tabel voorkomt wordt in de diverse boxen gekeken. De JOS-box moet hierbij op nr4 geplaatst zijn en behoeft niet te worden veranderd. Dit geldt ook voor de EDIT-, CALC-, en TOOLBUG-ROM. De betreffende statements komen voor in de SOS statement tabel. Alle andere boxen moeten het testbyte #E3 op #A001 hebben. Het terugkeer adres gaat naar #EFF5 in de DOS-rom. Hier in staat JMP#1803. Dit is gedaan omdat in de zoek-routines van boxen naar een hoge byte startadres wordt gekeken van #80 en hoger. Wanneer het statement ook niet in een der boxen voorkomt wordt gekeken of er statements op de geheugenkaart voorkomen. Indien hierin statements staan dient in #7FFD het testbyte #BF te staan en op #7FFE en #7FFF het start adres van de zoekroutine van deze statements. Het terugkeer adres van deze routine dient #1806 te zijn, waar een directe jump staat om de JOS-box voor te zetten, om tenslotte in de JOS-box te kijken.

Nu volgt nog de zoek-routine met een korte toelichting:

```
1838: LDA @#08      laadt accu.
183A: STA #0237     initialiseer uitlees-adres.
183D: DEC #0237     verlaag boxnummer.
1840: BMI #1859     kijk of alle boxen gepaseerd zijn.
1842: STA #0237     zet nummer in uitleesadres.
1845: STA #BFFF     sckakel de box voor.
1848: LDA #A000     laadt testbyte box.
184B: CMP @#40      controleer box.
184D: BNE #183D     geen statementbox ga naar de volgende box.
184F: LDA #A001     laadt tweede testbyte.
1852: CMP @#E3      controleer
1854: BNE #183D     geen juiste box ga naar de volgende.
1856: JMP #A002     juiste box ga naar de zoekroutine.
1859: LDA #7FFD     laadt testbyte geheugen-kaart.
185C: CMP @#BF      controleer.
185E: BNE #186A     niet de juiste ga kijken naar JOS-box.
1860: LDA #F77C     laadt tweede testbyte.
1863: CMP @#E3      controleer.
1865: BNE #186A     niet de juiste ga kijken naar JOS-box.
1867: JMP (#7FFE) ga naar zoekroutine op geheugen kaart.
186A: LDA @#04      laadt JOS-box nummer.
186C: STA #BFFF     zet hem voor.
186F: STA #0237     zet in uitleesadres.
1872: LDA #A001     laadt testbyte JOS-box.
1875: CMP @#BF      controleer.
1877: BNE #1883     geen JOS-box dan naar einde.
1879: LDA #A032     laadt tweede testbyte.
187C: CMP @#44      controleer.
187E: BNE #1883     geen JOS-box dan naar interpreter.
1880: JMP #A002     naar zoekroutine JOS-box.
1883: JMP #C558     naar interpreter en ERROR 94.
```

Wanneer U wacht met het inzenden van copy, totdat Uw programma's naar Uw mening beter zijn dan de reeds gepubliceerde, denkt U fout. Simpele programma's van 10 regels zijn soms interresanter dan ingewikkelde van 100 regels. De eerste soort wordt trouwens sneller ingetikt dan de tweede.


IC 10
74LS 244
IC 11
74LS 244
IC9
74 LS 245
N5
N6
N7
C11
100N
IC15
IC16
11x27k
8x27k
6116
IC14
74LS138
enable
R/C
ϕ2
dir
D1
1N4001
D2
LED
R29
R30
R31
Power down
3V
PD
computer aan
uit
N 1,2,4,5 = IC12 - 74L
N 3,6,7 = IC13 - 74L
N 8,9,11,13 = IC10 - 40
N 10,12,14,15 = IC16 - 40


SCHEMA 16 K RAM KAART
Fed. Acorn Computer-
clubs Nederland
(. van Drunen '83)
IC 2
IC 3
IC 4
IC 5
IC 6
IC 7
IC 8
IC 15 + IC 16 [4071]
N8
N9
N10
N12
N13
N14
N1
N2
N3
N4
74LS138
I
II

```
 10    REM ZEROPAGE RELOCATOR
 20    REM DOOR C.KWAKERNAAK
 30    REM DIT PROGRAMMA ZOEKT EN
 40    REM VERANDERT ZERO-PAGE ADRESSEN,
 50    REM IN EEN GEGEVEN PROGRAMMA(DEEL).
 60    REM (GEBRUIKT DE JOBBOX)
 70    DIM A(256),C(256),S(32),LL(1),P(-1)
 80    F.I=0TO256;A?I=I+1;C?I=I;N.
 90    W=!#208&#FFFF;P.$21;LL1=P
100  [:LL1STA#80;TXA;PHA;TYA;PHA
110    LDA#81;TAX;INC#81;LDA#80;STAS,X
120    PLA;TAY;PLA;TAX;LDA#80;JMP W
130  ];P.$6$12;G.b
140  s REM VANG DISAS.STRING AF
150    P.$21;?#208=LL1;?#209=LL1/256
160    ?#81=0;?#90=0;?#92=O;?#93=O/256
170    LI.#A7E6;P.';REM DISAS (#92,#93)
180    ?#208=W;?#209=W/256;S?((?#81)-2)=13
190    P.$6;R.
200  b IN."START ADRES"B,"EIND ADRES"E
210    P.''"HET PROGRAMMA ZOEKT NU ZERO-"'
220    P."PAGE INSTRUCTIES EN DRUKT ZE AF."''
230    O=B;I=0;DO GOS.s;L=?#91;Y=-1
240    IF S?20=CH"?";G.e
250    IF L=2;IF S?20=CH"B";IF S?21=CH"I";G.g
260    IF S?20=CH"B";G.e
270    IF S?24=CH"@";G.e
280  g IF L=2;Y=O?1;IF S?30=CH"Y";A?Y=Y
290    IF L=3;IF O?2=0;Y=O?1
300  e IF Y<>-1;P.&O"    "$(S+20)';A?Y=Y
310    O=O+L;U.O>=E
320    P.$12;F.I=1TO16;P.';N.;P.$#E
330    P."DE GEBRUIKTE ZP ADRESSEN ZIJN,"'
340    P."(AANEEN VOLGENDE ADRESSEN STAAN"'
350    P."OP DE ZELFDE REGEL)"'
360    @=3;P.":";F.I=0TO255
370    IF A?I=I;P.&I;IF A?(I+1)<>(I+1);P.'":"
380    N.;P.$#F
390    P."TYP NU DE VERVANGERS IN"'"I=IDEM"'
400    F.I=0TO255
410    IF A?I=I;P."#"&A?I" DOOR ";IN.K;C?I=K;P.$#B$#D
420    N.
430    P.'"NU WORDEN DE ADRESSEN VERVANGEN"''
440    O=B;I=0;DO GOS.s;L=?#91;Y=-1
450    IF S?20=CH"?";G.h
460    IF L=2;IF S?20=CH"B";IF S?21=CH"I";G.i
470    IF S?20=CH"B";G.h
480    IF S?24=CH"@";G.h
490  i IF L=2;Y=O?1
500    IF L=3;IF O?2=0;Y=O?1
510  h IF Y<>-1;O?1=C?Y
520    O=O+L;U.O>=E;P."KLAAR"';E.
```

Het onderstaand artikel beschrijft de bouw en werking van een modulair interface systeem voor de Acorn Atom.
De UI (Universeel Interface) is ontwikkeld binnen de vakgroep Anorganische Chemie van de R.U. Groningen.
Deze beschrijving komt in grote lijnen overeen met het artikel:""UI" van Columbus" uit Elektronica nr.10 1982 van uitgever Kluwer Technische Tijdschriften B.V.. Auteur van het betreffende artikel is de heer H.C.G. Druiven van de R.U. Groningen.
De tekst is hier en daar wat ingekort en aangepast voor de Acorn Atom.

Een van de belangrijkste voordelen van dit ontwerp is,dat wanneer men overschakelt naar een ander computersysteem niet alle, vaak dure of met moeite ontwikkelde,interfaces dienen te worden vervangen.
De UI heeft met andere interface systemen, zoals CAMAC en IEEE 488, gemeen, dat data-uitwisseling met de computer via in-en output wordt verzorgd. Dit heeft als voordeel,dat de interfacebus optimaal aan het interface-gebeuren kan worden aangepast en volledig computer-onafhankelijk kan worden uitgevoerd.Een nadeel is,dat altijd meer instructies nodig zijn om dezelfde informatie uit te wisselen dan bij een interface direct op de computerbus. Deze meer-instructies dragen echter wel bij tot de duidelijkheid van de te gebruiken software.

De UI is speciaal ontworpen voor het gebruik met 8-bit's microcomputers. Doordat de UI van een gescheiden in-en output bus is voorsien,kan deze bijna geheel statisch worden gebruikt en met behulp van een eenvoudige 10 Mhz scoop op zijn juiste werking worden gecontroleerd. Omdat de UI-bus ook veel overeenkomst vertoont met een normale computerbus kan dezelfde ontwerpfilosofie worden toegepast.
De UI is opgebouwd binnen de behuizing van 19-inch kast,waarin eurokaarten met een frontbreedte van één inch (of meervouden daarvan) kunnen worden gestoken. De kaarten zijn voorzien van een 31-polige connector.
Er is vanaf het begin van de ontwikkeling getracht een logisch verband te leggen tussen de hardware en de te gebruiken software. Zo is er direct verband tussen de plaats van de interfacekaarten en het te selecteren kaartadres;de eerste kaart op slot 1, en kaart 12 op slot 12. Om op één kaart verscheidene functies te kunnen laten uitvoeren, is voorzien in een adresbus met een breedte van 4-bit, waarmee voor in-en output ieder 16 functies (0 t/m 15) kunnen worden geselecteerd. Alleen adres 15 is gereserveerd voor interrupt-doeleinden. Van de overige lijnen van de bus zijn 18 lijnen voor in en output, twee voor interrupt en vier voor de benodigde voedingsspanningen.
Schematisch komt de Atom uitvoering van de UI-Bus er uit te zien als in fig.1. Dit schema wijkt enigszins af van het oorspronkelijke schema. Dit komt,doordat de Atom nog niet beschikt over vier vrije I/O-poorten,zoals bijvoorbeeld de Aplle II. Deze vier I/O-poorten zijn nog als Extra in de UI-Bus opgenomen ook de Atom geschikt te maken.
Het interface gedeelte tussen de Atom en de UI-Bus bestaat uit twee printen, in fig.1 aangegeven met Card1 en Card2. Deze twee Card's worden niet, zoals in het oorspronkelijke ontwerp,voor in de rack geplaatst,maar achter in de kast van de UI-bus.

ADRES-INTERFACEKAART (card 1)

Deze kaart verzorgt de cardselect-signalen (slot 1 t/m 12) en het 4-bit adres (zie fig.3).
Op de card bevindt zich een PIA van het type 6820/6520.Beide poorten van de PIA zijn geprogrammeerd als output. Poort-A levert de 4 adres lijnen voor de Bus. Deze worden gebufferd m.b.v. een LS 373. Poort-B levert 4 lijnen,die door een decoder 74154 omgezet worden in 16 cardselect signalen, hiervan worden er slechts 12 gebruikt(Q1 t/m Q12). Zowel Card 1 als Card 2 worden via één kabel op plug 8 aangesloten(hierover later meer). Voor de noodzakelijke adresdecodering van de beide PIA's zorgt de LS 139 op Card 1. Deze haalt uit de lijnen A2,A3,NB400 de benodigde chipselect-signalen.Hierbij blijven 2 signalen ongebruikt n.m.l. Q2 en Q3 van de LS 139, deze kunnen eventueel nog voor andere uitbreidingen gebruikt worden.

I/O-INTERFACEKAART (card 2)

Deze kaart verzorgt de in-en output Bus en de interrupt van de UI-Bus en voorziet tevens de reset lijn van power-on-reset (zie fig.3).
Ook op deze card bevindt zich een PIA van het type 6820/6520. Poort-A is nu echter geprogrammeerd,als input en poort-B als output. Beide poorten worden weer gebufferd door een LS373. Verder bevinden zich op deze kaart een strobe,enable,interrupt en reset-lijn.

Technische gegevens:
De input bus van de I/O-interfacekaart wordt met maximaal 10 TTL-of 40 LS TTL-poorten per lijn belast en de enable-lijn mag met niet meer dan 1 TTL-poort worden belast en moet met 470 ohm aan de +5 volt worden gelegd.
De output bus moet in staat zijn om per lijn 13 TTL of 48 LS-TTL poorten aan te sturen (de LS 373).
Voor de overige interfaces geldt dat de output bus met niet meer dan 1 TTL of 4 LS-TTL-poorten per lijn wordt belast. De input bus van de overige interfaces moet in staat zijn om 10 TTL-of 40 LS TTL-poorten aan te sturen (normale TTL output). De enable lijn mag met een open collector worden aangestuurd. Verder geldt,dat cardselect met 10 TTL-of 40 LS TTL-poorten mag worden belast,adres met niet meer dan 1 TTL-of 4 LS TTL-poorten per lijn. IRQ IN wordt met niet meer dan 10 TTL-poorten belast, terwijl IRQ OUT in staat moet zijn om 10 TTL-poorten aan te sturen.

INTERFACELIJNEN

De functies van de afzonderlijke lijnen zijn als volgt:

1. cardselect
Deze lijn geeft aan of de desbetreffende kaart wel of niet geselekteerd is. De lijn voert negatieve logika, zodat een onderbreking van deze lijn gezien wordt als "not select".Dit voorkomt in de meeste gevallen van een defect, dat twee of meer kaarten tegelijk op de bus toegelaten worden.

Card select

ADRES

CARD 1

5V + AARDE

PL-8 5V CS1

1 2 3 4 5 6 7 8 9 10 11 12

5V/5A

±15V 100mA + AARDE

CARD 2

IN

OUT

CONTROL

PL-8

Verbindings kabel
UI-BUS ATOM.

achterkant kast Atom

25-Polige D-Connector
25-polig Chassis-deel.

20 aderige Flatcable

PLUG-8

Spectra header.

Side 2. PCB.

fig.1

Structuur v.d UI-Bus voor de Atom

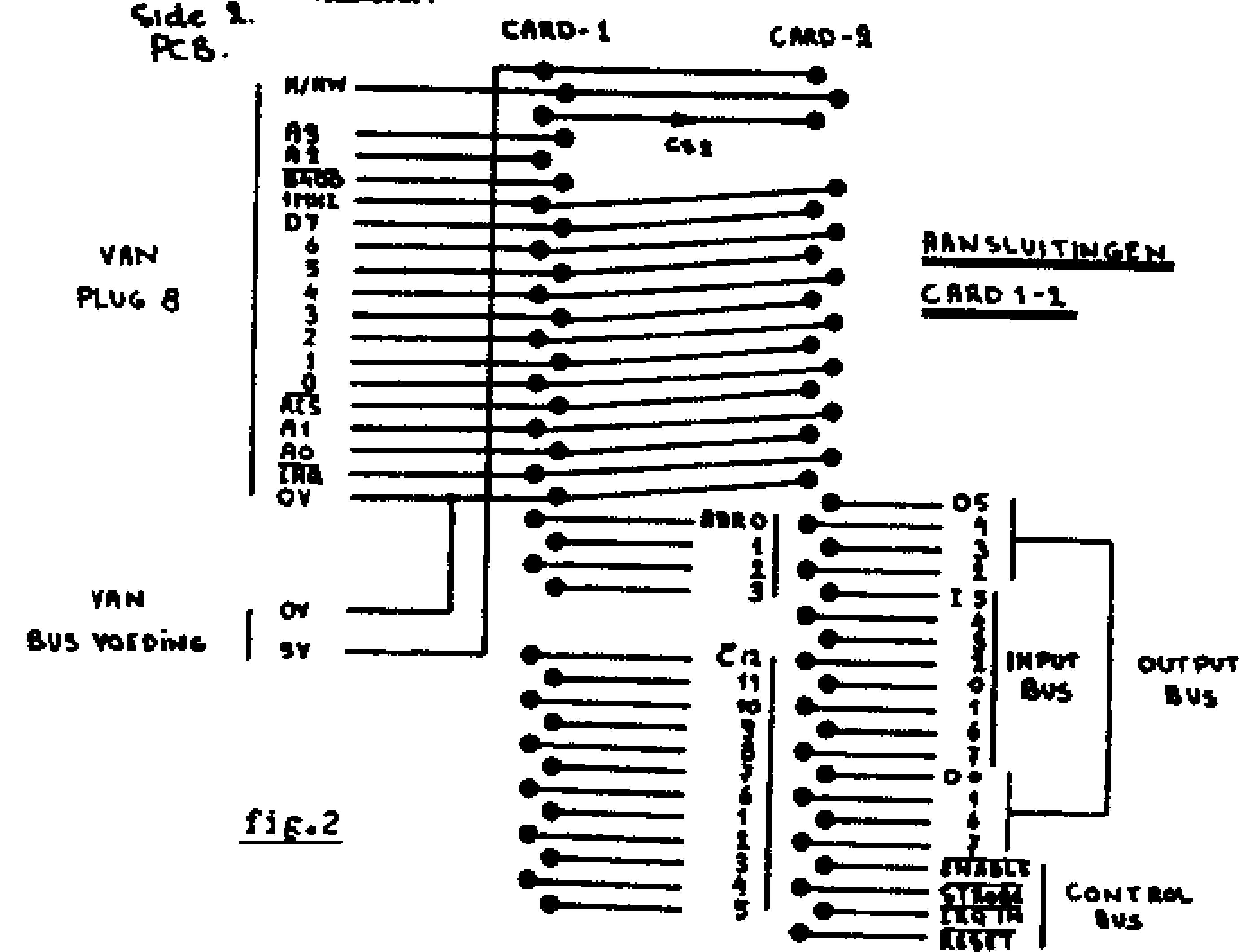


fig.2

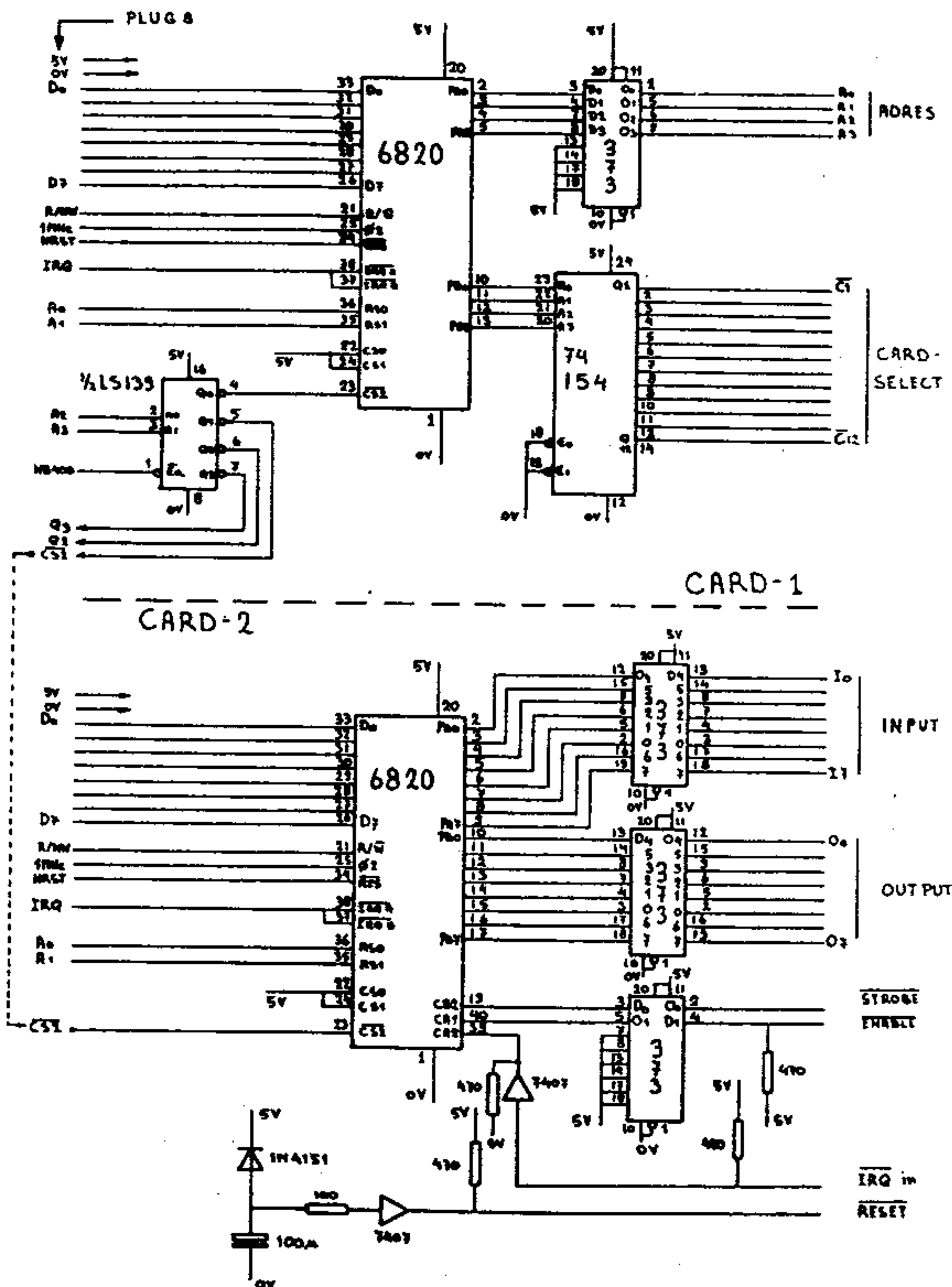

PLUG 8
6820
373
74
154
ADRES
CARD-SELECT
CARD-1
CARD-2
6820
INPUT
OUT PUT
STROBE
ENABLE
IRQ in
RESET
1N4151

Fig.3

2 t/m 5. adres
Er zijn vier adres-lijnen, waardoor per kaart uit zestien verschillende functies kan worden gekozen. Een adres kan zowel voor in- als output dienst doen, zodat feitelijk 32 functies beschikbaar zijn. Adres 15 is gereserveerd voor interupt doeleinden.

6 t/m 14. output data bus
Deze bus gaat van computer naar randapparaat en voert positieve logika met uitzondering van de strobe-lijn op 14. Voor de negatieve logika van deze lijn gelden min of meer dezelfde overwegingen als bij cardselect. Strobe dient minstens gedurende 500 ns laag te zijn.

15. Interupt request in ($\overline{\text{IRQ IN}}$)
Op deze lijn verschijnt de IRQ van de hoger gelegen kaarten. De IRQ op deze lijn wordt aan IRQ OUT doorgegeven, als de kaart op dat moment geen IRQ behandelt. Vraagt de kaart zelf wel een IRQ aan, dan wordt de IRQ van de hoger gelegen kaarten onderbroken, totdat de IRQ van de kaart zelf volledig is afgehandeld. Dit vereenvoudigt de te gebruiken software en bepaalt tevens de prioriteit van de aangevraagde interrupt. De prioriteit loopt van kaart 1 tot 12. Mocht tijdens de ontwikkeling van een systeem de prioriteit tussen twee kaarten wijzigen, dan hoeven de kaarten slechts van plaats te verwisselen. Op de kaart moet een voorziening worden getroffen, die de IRQ status flag automatisch reset, nadat adres 15 aan deze kaart is aangeboden. De IRQ van hoger gelegen kaarten mag pas weer worden doorgegeven, als een output strobe op adres 15 van de desbetreffende kaart wordt gegeven.

16. Interrupt request out ($\overline{\text{IRQ OUT}}$)
Op deze lijn verschijnt de interrupt-aanvraag van de desbetreffende en hoger gelegen kaart. Zie hierover IRQ IN. Beschikt de kaart zelf niet over een interrupt mogelijkheid, dan dient lijn 15 met 16 te zijn doorverbonden.

17t/m 25. Input databus
Deze bus gaat van randapparaat naar computer. De informatie heeft positieve logika en wordt door de computer ingelezen, nadat lijn 25 (enable) laag is geworden. Voor de negatieve logika van deze lijn gelden dezelfde overwegingen als bij cardselect.

26. $\overline{\text{reset}}$
Deze lijn is via een weerstand van 470 ohm met de +5 volt verbonden en wordt gedurende 10 ms laag gehouden, als de kast van netspanning (of andersoortige voeding) wordt voorzien.

27. NC

28. +5 volt
De +5 volt voeding binnen de kast dient deze lijn van minstens 2 ampère te voorzien. Per kaart is dan 133 mA beschikbaar. (eventueel kan men deze spanning van de atom voeding betrekken, wanneer deze zwaar genoeg is)

29. -15 volt
Deze lijn wordt met minstens 100 mA gevoed, zodat per kaart 6,6 mA beschikbaar is.

30. +15 volt
Idem als bij 29.

31. GND
De nul-lijn bestaat uit minstens 1 mm^2 koperdraad en is doorverbonden met de diverse voedingen en de nul van de Atom via de kabel op plug 8.
De nul mag niet worden doorverbonden met de randaarde van het lichtnet.

In de kast kunnen ook kaarten met alleen analoge functies worden geschoven. Van die kaarten dient pen 15 met 16 te zijn doorverbonden. Fig.4 geeft de maten van de diverse kaarten weer.
Achter in de kast bevindt zich vodoende ruimte om de beide interface kaarten met de Atom en de voedingsprint en trafo onder te brengen.
De nul van de voedingen is niet verbonden met de randaarde van het lichtnet om ongewenste aardlussen te voorkomen. Wel dient de metalen kast met de randaarde te zijn doorverbonden.

1. cardselect		
2 A0	adres	
3 A1		
4 A2		
5 A3		
6 O0	output bus data van computer naar randapparatuur	
7 O1		
8 O2		
9 O3		
10 O4		
11 O5		
12 O6		
13 O7		
14 strobe		
15 IRQ IN		
16 IRQ OUT	(niet voor I/O-interface)	
17 I0	input bus data van randapparaat naar computer	
18 I1		
19 I2		
20 I3		
21 I4		
22 I5		
23 I6		
24 I7		
25 enable		
26 reset		
27 NC		
28 +5 volt		
29 −15 volt		
30 +15 volt		
31 GND		

Tabel 1 Connectoraansluitingen van de overige interfacekaarten.

DE INTERFACE/PRINTEN (card 1 en 2)

De print lay-out van card 1 is gegeven in fig. 5, en is enkelzijdig.
De gebruikte connector is 40-polig met een contact-afstand van 0,1 inch, en dubbelzijdige vorkkontakten. De door mij gebruikte connectoren waren afkomstig uit een oude telefoon-modem, waarvan overigens ook de kast voor de UI werd gebruikt, dit is de reden waarom de contacten op de print op een plaats verspringen. In een van de lege kamers van de connector werd een stukje print plaat gelijmd, op de overeenkomstige plaats werd uit de print plaat een stukje gezaagd, zodat het nu onmogelijk wordt de print in de verkeerde connector te steken of verkeerd om in te steken.
Fig. 6 geeft de componenten zijde van card 1 (let op de draadbrug onder de 6820).
De print lay-out van card 2 is dubbelzijdig. Fig.7 geeft de koperzijde en fig.8 de koperbanen aan de componentenzijde. Alle punten waar doorgemetaliseerd moet worden, bevinden zich bij de aansluitdraden van de passieve componenten en de IC's. Door voor de IC voetjes gebruik te maken van IC-connectorstrip, kan men deze gemakkelijk aan weerszijden van de print vastsolderen. Voor de connector geldt hetzelfde als voor de connector van card 1.
De componenten opstelling van card 2 is gegeven in figuur 9.
De condensatoren van 100 nF dienen ter ontkoppeling van de voedingsspanning.

Voor de bedrading tussen de printen de bus en plug 8 zie fig.2 (let op de CS 2 lijn van card 1 naar 2)

componentenzijde

Fig. 4. Afmetingen van de kaarten en connectoren (bij benadering).

fig.4

16602 ATOM UI-BUS CARD-1

fig.5

LS 373

LS 139

100nF

154

6820

100 nF

fig.6

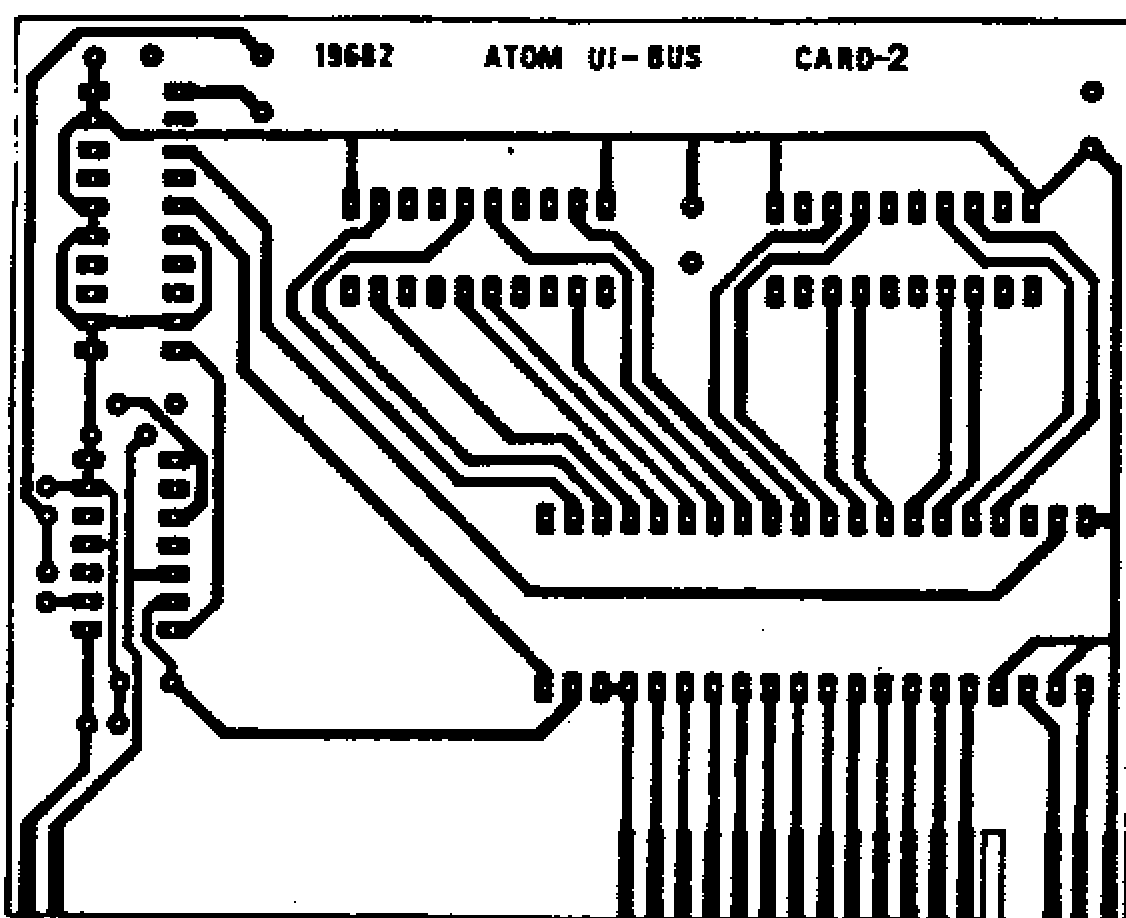


fig.7

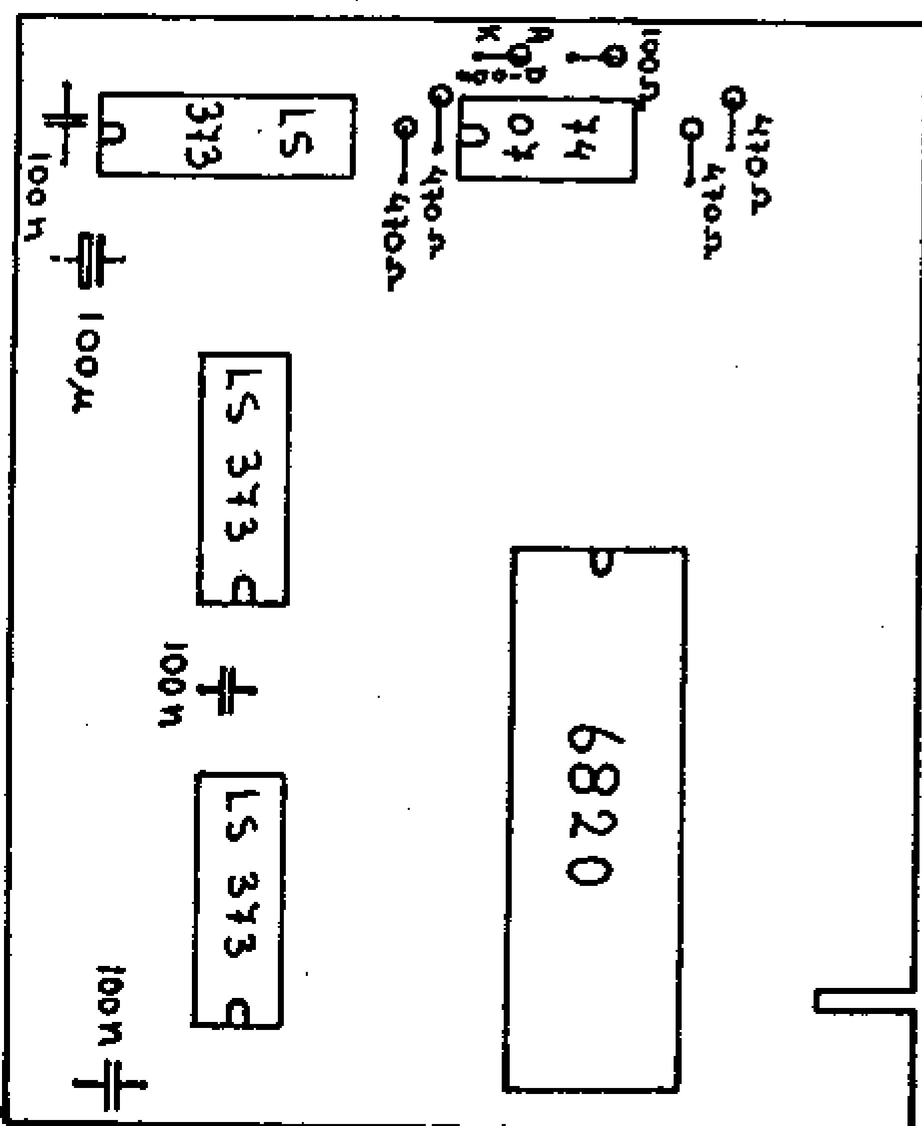


fig.9

fig.8

HET AANSLUITEN OP PLUG 8

Om de zaak demontabel te houden heb ik voor de volgende oplossing gekozen. Op zijde 2 van de Atom print heb ik in de gaatjes van plug 8 twee rijen rechte wafers (printstekerdeel) gesoldeerd. Vandaaruit gaat men dan verder met een haakse spectra-header en een stukje flatcable van ca. 20 cm naar de achterkant van de Atom-kast. Hierin plaatst men dan een 25-polige D-connector (chassisdeel). Op deze connector kan men dan de kabel,afkomstig van de Ul-bus,aansluiten.Let er wel op,dat geen draden verwisseld worden. De +5 v van plug 8 mag nergens op aangesloten worden.
Maak de kabel tussen de Atom en de UI-bus niet langer,dan absoluut noodzakelijk.

aansluitingen van plug 8 :

20	$\overline{RST}$	19	$\overline{IRQ}$
18	$R/\overline{W}$	17	1MHz
16	A2	15	A3
14	A0	13	D7
12	A1	11	0 volt
10	+5 volt (los)	9	$\overline{NMI}$ (los)
8	D0	7	D3
6	D2	5	D1
4	$\overline{B400}$	3	D4
2	D5	1	D6

Link 3 moet gemaakt worden om interrupt van de bus mogelijk te maken.

GEBRUIK VAN DE UI-BUS

1. Selecteer de juiste kaart en het juiste adres, waarbij de volgorde er niet toe doet.
2. Bij output: Zet de juiste informatie op de output databus en geef gelijktijdig, of even later, een 0,5 µs (of langer) strobepuls. De informatie op de output databus blijft staan tot nieuwe informatie wordt aangeboden.

   Bij input: (afhankelijk van het type interface)
   a. De toestand van de enable-lijn wordt genegeerd,zodat informatie op de input databus direct door de computer kan worden ingelezen. In deze gevallen wordt de enable lijn door cardselect aangestuurd.
   b. De informatie op de input databus wordt door de computer overgenomen,nadat de enable-lijn laag is geworden. In deze gevallen dient de interface zodanig te zijn geconstrueerd, dat zowel input data als enable "waar" blijven,totdat en/of een nieuwe kaart wordt geselecteerd.

en/of een nieuw adres wordt geselecteerd
en/of een output strobe wordt gegeven.

INITIALISATIE VAN BEIDE PIA'S

De beide PIA's bevinden zich in het adres bereik van ≠B400 t/m ≠B407.

≠B400 Data Dir. Reg.,Port A (adres poort)
≠B401 Control Reg. A

≠B402 Data Dir. Reg.,Port B (cardselect)
≠B403 Control Reg. B

≠B404 Data Dir. Reg.,Port A (input poort)
≠B405 Control Reg. A

≠B406 Data Dir. Reg.,Port B (output poort)
≠B407 Control Reg. B

Als bit 2 van het control register 0 is krijgt men toegang tot het Data Dir. Reg. en als bit 2 1 is krijgt men toegang tot de bijbehorende I/O-poort.
De verdere functie van het control register komt in een volgend artikel aan de orde.

testroutine:

```
10 P=≠B400; REM basis adres
20 P?1=0;P?3=0;P?5=0;P?7=0; REM maak alle stuurbits nul en geef toegang
                                tot alle Data Dir. Registers.
30 P?0= ≠FF; REM maak alle bits van poort A van PIA 1 output
40 P?2= ≠FF; REM maak alle bits van poort B van PIA 1 output
50 P?4= 0  ; REM maak alle bits van poort A van PIA 2 input
60 P?6= ≠FF; REM maak alle bits van poort B van PIA 2 output
70 P?1=4;P?3=4;P?5=4;P?7=4; REM maak stuurbit 2 1 en geef toegang tot alle
                                I/O-poorten
1001IN."CARD" C; P?2=C
110 IN."ADRES" A; P?0=A
120 IN."OUTPUT" O; P?6=O
130 I=P?4; PRINT"INPUT=" I'
140 GOTO 1
```

Bovenstaand programma initialiseert eerst alle PIA registers en komt daarna in een lus,die data vraagt en deze daarna op de bus zet,verder worden de data uitgeprint,die op dat moment op de input bus staan.

Door de vrijwel statische werking van de UI-bus is het mogelijk de juiste werking stap voor stap te volgen. De display-kaart,zoals hierna beschreven,is daarbij een handig hulpmiddel. De LED's geven direct de waarde van in- en output weer. Het vooraf selecteren van deze kaart is niet nodig. Het print-ontwerp van deze kaart staat helaas nog niet op papier,maar zal in een volgend nummer komen. Voor het schema zie fig.10.

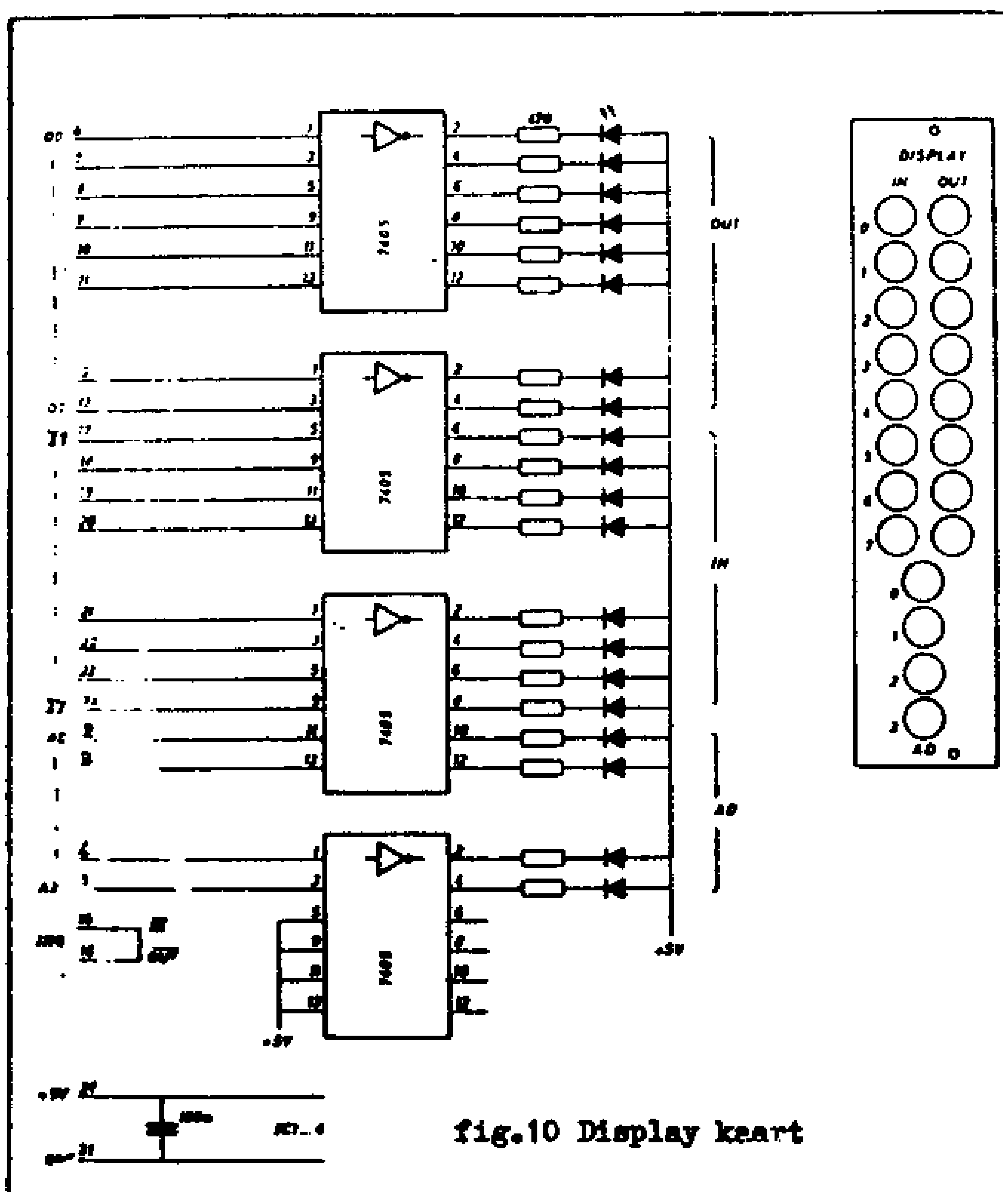


fig.10 Display kaart

TIMING

Om interface-kaarten van de ene gebruiker uitwisselbaar te maken met die van een ander, dienen nog wat afspraken te worden gemaakt over de timing van de verschillende signalen. De timing is zodanig, dat geen last wordt ondervonden van bedradingscapaciteiten tot 500pF en eventuele spike's t.g.v. adresdecodering.
Cardselect, adres en output mogen gelijk of na elkaar worden aangeboden. Strobe verschijnt gelijk, of na output, terwijl cardselect, adres en output niet veranderen gedurende strobe en 500nS daarna. Output wordt op de positieve flank van strobe door de interface overgenomen en blijft geldig tot nieuwe output data wordt aangeboden.
Afhankelijk van het type interface wordt:

a. de toestand van de enable-lijn genegeerd. In dit geval kan de enable-lijn direct via een open-collector uitgang door cardselect worden aangestuurd.

b. input data gelijk of eerder dan het verschijnen van enable geldig en wordt niet eerder dan 500nS na het verschijnen van enable door de computer ingelezen. In dit geval blijven input data en enable waar tot:
   en/of cardselect verandert
   en/of adres verandert
   en/of een output strobe verschijnt.

Alle signaalveranderingen vinden binnen 250 nS plaats en alle lijnen zijn <0,8 volt of >2,5 volt.

DE VOEDING

Voor de 5 volt voeding kan men b.v. gebruik maken van een LM 323(3 amp) of, zoals ik zelf heb gedaan, de twee stabilisatoren van 1 amp, die in de Atom overbodig waren geworden nemen, en dan elk van de stabilisatoren een helft van de bus laten voeden. Voor de +15 en -15 volt kan men resp. een 78L15 en een 79L15 nemen (100mA).

INSTEEKKAARTEN

Momenteel heb ik drie kaarten gebouwd en getest n.m.l. de displaykaart uit Elektronica een eprom programmer-kaart en een A/D-kaart(voltmeter).
Verder zijn er plannen voor een D/A-kaart en een kaart voor een cassetterecorder besturing(recorder met servo-bediening).
Hierbij voeg ik nog twee schema's uit Elektronica, die ook als men ze niet nabouwt interessant om te bestuderen zijn.
In een volgend nummer hoop ik te beginnen de beschrijving van de A/D-kaart, die ook als men de UI-bus niet bouw, door enkele kleine wijzigingen op de Atom kan aansluiten.

Verder kan ik melden, dat er reeds een firma is die de UI-bus in de handel brengt en dat men ook aan een Atom interface werkt.

Dit artikel kwam tot stand met de welwillende toestemming van de auteur; de heer H.C.G. Druiven, R.U. Groningen en de Redaktie van Elektronica, Kluwer Technische Tijdschriften.

W.Kuijpers

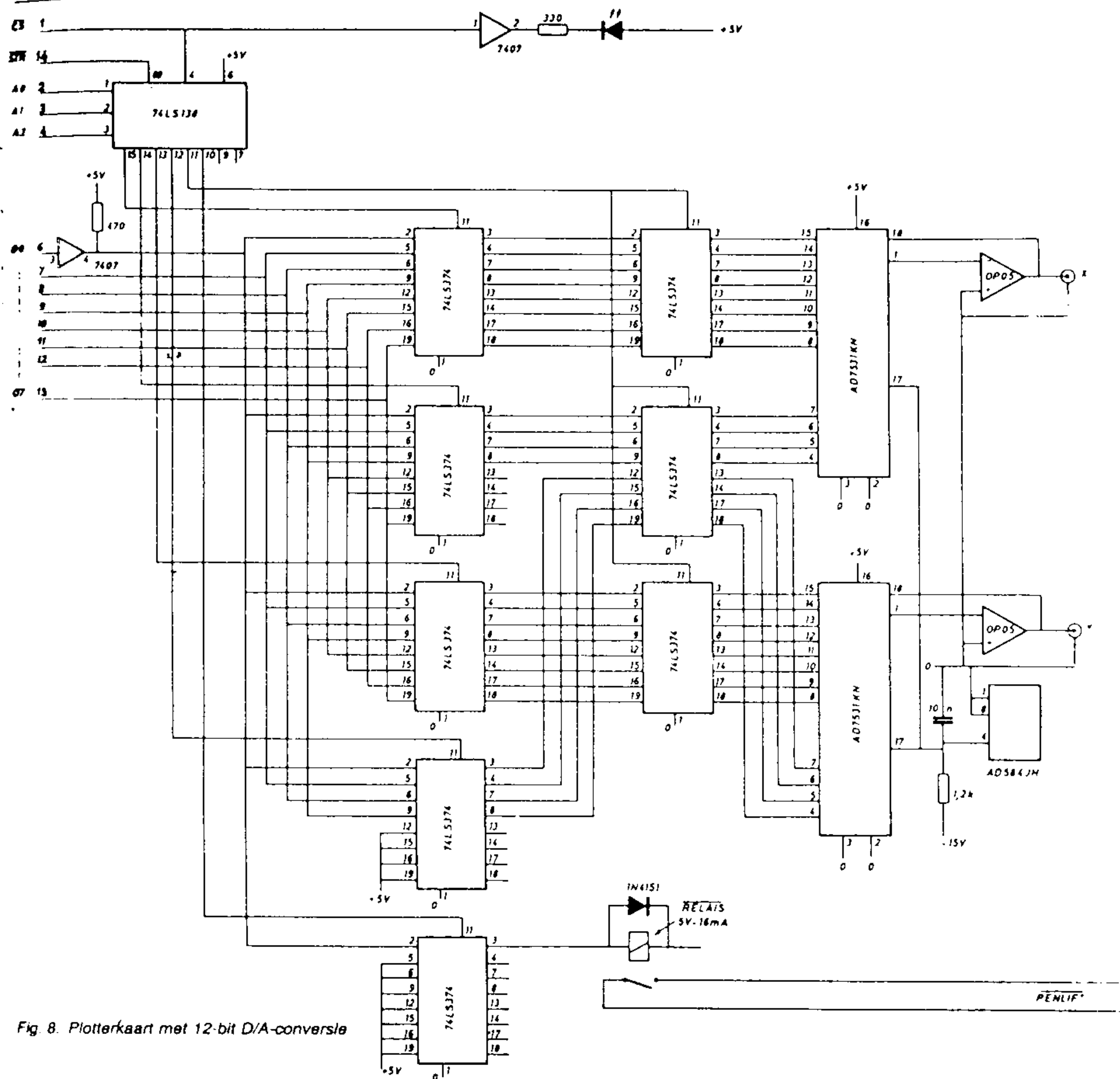


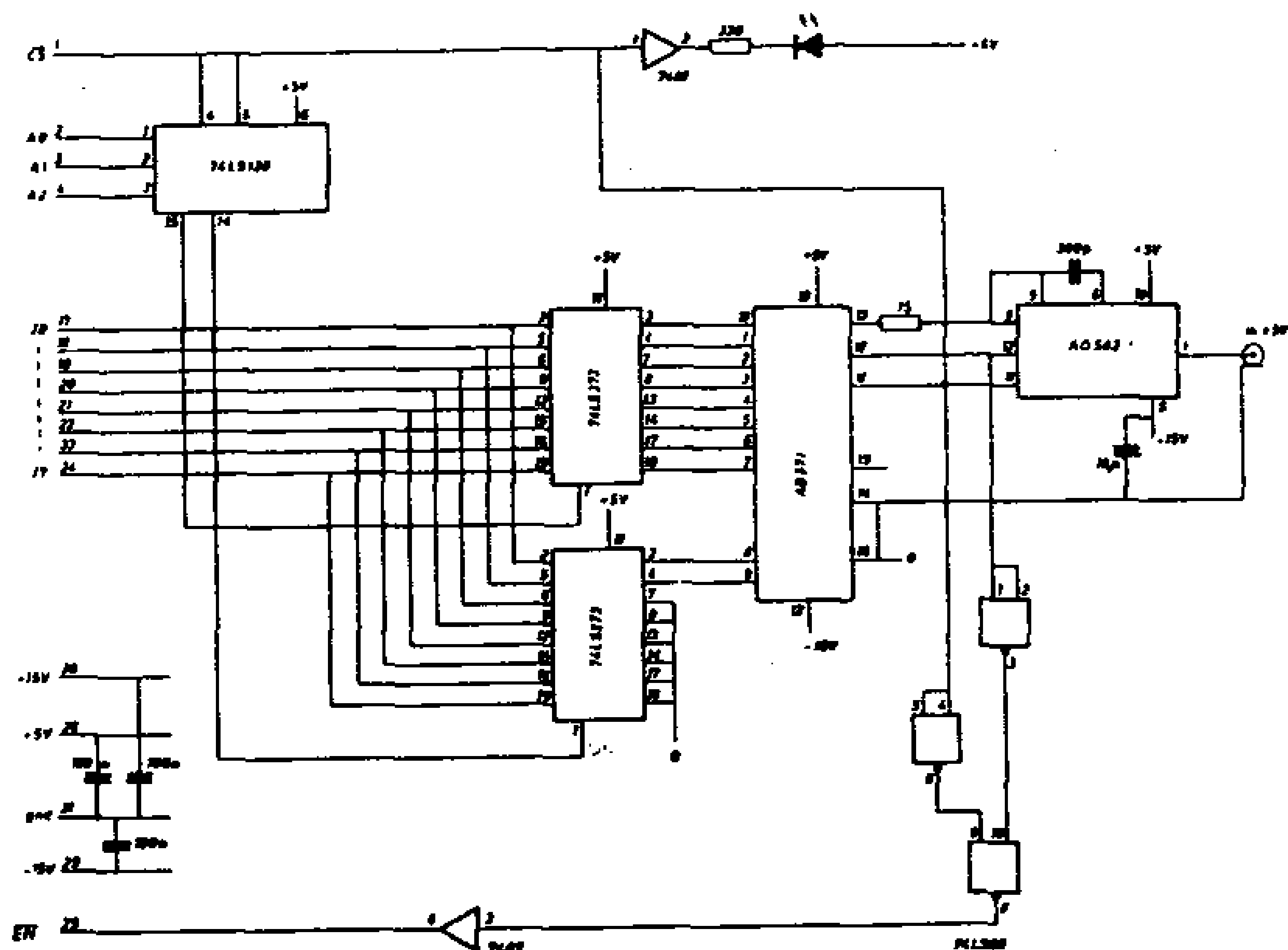
Fig. 8. Plotterkaart met 12-bit D/A-conversie

Afb. 9. Subroutine in Apple II BASIC voor het aansturen van de plotter.

```
10000 REM X EN Y OUT
10100 POKE CA,12
10200 IF X<0 THEN XR=0:GOTO 10300
10210 IF X>4095 THEN XR=4095:GOTO 10300
10220 XR=INT(X+.5)
10300 IF Y<0 THEN YR=0:GOTO 10400
10310 IF Y>4095 THEN YR=4095:GOTO 10400
10320 YR=INT(Y+.5)
10400 XH=INT(XR/256)
10410 XL=XR-256*XH
10450 YH=INT(YR/256)
10460 YL=YR-256*YH
10500 POKE AD,0:POKE OUT,XL
10510 POKE AD,1:POKE OUT,XH
10520 POKE AD,2:POKE OUT,YL
10530 POKE AD,3:POKE OUT,YH
10590 POKE AD,4:POKE OUT,YH
      :REM X en Y NAAR D/A
10900 RETURN
```

Afb. 10. Subroutines voor „pen up" en „pen down".

```
11000 REM PEN UP
11100 POKE CA,12
11200 POKE AD,5:POKE OUT,0
11900 RETURN
12000 REM PEN DOWN
12100 POKE CA,12
12200 POKE AD,5:POKE OUT,1
12900 RETURN
```




```
13000 REM A/D CONVERTER
13100 POKE CA,11
13200 IF PEEK(CA(2))<128 THEN 13200:REM WAIT FOR ENABLE
13300 POKE AD,0:BP=PEEK(IN)
13310 POKE AD,1:BP=BP+256*PEEK(IN)
13900 RETURN
```

Afb. 12. Bij de A/D-converterkaart behorende BASIC-routine.

Duitsland 3 NDR. iedere maandag om 1900 MIKRO-ELEKTRONIKA.

INLEIDING:
Met de tekstverwerker kunnen documenten zoals brieven, folders of boeken gemaakt worden. De tekst kan achter elkaar door ingevoerd worden en hoeft niet in zinnen afgebroken te worden. Dit doet de EDITOR automatisch.
In de ATOM met geheugen van #2800 tot #3C00 kunnen ongeveer 5000 tekens bewerkt worden, wat overeenkomt met ongeveer 3 pagina's A4 tekst. Er is ook een zgn. correctiebuffer, een geheugengedeelte waarin tijdelijk een stuk tekst kan worden opgeslagen om het later te gebruiken.(max. 448 tekens)

AANZETTEN EDITOR:
Het aanzetten van de editor gebeurt door EDIT te tikken en daarna op de toets RETURN te drukken.
Als er geen floating point ROM in de ATOM aanwezig is, moet de editor aangezet worden met LINK #ACFD, RETURN.
Op het scherm verschijnt de vraag:"OLD TEXT?".
Als er geen tekst in het geheugen zit, die bewerkt moet worden, kan er op RETURN gedrukt worden. Daarna wordt het scherm leeggemaakt en verschijnt er een wit blokje dat het "einde tekst" aangeeft. Op de onderste regel verschijnt het nummer van de "beeldscherm-pagina" met daarachter de zgn. promt die aangeeft dat de editer voor gebruik gereed is. Dit ziet er als volgt uit: p01>

INTIKKEN TEKST:
Na intikken van het commando E verschijnt op de onderste regel achter de de promt E:
Het geheel gaat er dan als volgt uitzien: p01>E:
Nu kan er tekst ingetikt worden. Dit kan achter elkaar door gebeuren en de tekst hoeft niet in zinnen afgebroken te worden. Dit doet de EDITOR automatisch. Een nieuwe regel kan begonnen worden met de toets RETURN.
Een regel overslaan wordt gedaan door een lege regel te maken (2*RETURN).
Als een woord niet op de regel past hoeft het NIET te wòrden afgebroken, maar kan op de volgende regel doorgetikt worden. Foute letters kunnen met de toets DELETE teruggenomen worden.
Het intikken van tekst dan beeindigd worden door op de toets COPY te drukken. Hierna wordt de laatste schermpagina tekst getoond.

BEKIJKEN INGETIKTE TEKST:
Voor het bekijken van tekst zijn de volgende commando's beschikbaar:
S (start of text). Dit toont de eerste schermpagina vanaf het begin v/d tekst.
Z (end of text). Toont de laatste schermpagina tot het einde van de tekst.
N (next page). Toont een volgende schermpagina.
P (previous page). Toont de voorafgaande schermpagina.
↑ (roll up a line)Schuift de tekst 1 regel,omhoog.

CURSORBESTURING:
De op het scherm getoonde tekst kan nu gewijzigd of verbeterd worden. Hiervoor wordt de cursor (het liggend streepje onder de regel) gebruikt. De cursor kan worden bestuurd met de volgende toetsen:
Het liggende dubbele pijltje ↔ stuurt de cursor naar rechts, met SHIFT tegelijk ingedrukt naar links.
Het staande dubbele pijltje ↕ stuurt de cursor omhoog, samen met SHIFT omlaag.
De cursor blijft automatisch doorlopen als de toets ingedrukt blijft. Dit maakt een snelle positionering mogelijk.
Met [verspringt de cursor naar het begin en met] naar het einde van de regel waarin hij zich bevindt.

Met H verspringt de cursor weer naar zijn beginpositie linksboven op het scherm.
De plaats waar de cursor zich bevindt kan "gemarkeerd" worden door op de toets @ te drukken.
Dit wordt aangegeven met een @ op de onderste regel. Het gemarkeerde punt kan worden gewist met de toets ESC.
Als daarna de cursor elders in de tekst geplaatst wordt kan ALLE tekst tussen @en de cursor met bepaalde commando's gewijzigd worden. Als er geen positie gemarkeerd is met @ wordt alleen de plaats van de cursor beinvloed.

WIJZIGEN OF CORRIGEREN:
Voor het wijzigen of corrigeren van de tekst zijn de volgende commando's beschikbaar:
A - Met A kan tekst tussengevoegd worden achter de huidige plaats van de cursor. Na het tikken van A verschijnt op de onderste regel A: waarachter de tekst getikt kan worden. Met COPY wordt de tekst tussengevoegd dan.
B - Met B wordt op dezelfde manier tekst tussengevoegd VOOR de cursor.
E - Met E kan tekst worden toegevoegd achter het einde van de tekst, op de plaats van het witte blokje.
DELETE - Als op de knop DELETE gedrukt wordt, dan wordt alle tekst tussen @ en de cursor gewist. Als er geen positie gemarkeerd is met @ wordt alleen de plaats van de cursor gewist.
C- Als op C gedrukt wordt, dan wordt de tekst tussen @ en de cursor gekopieetd in de correctiebuffer. Deze tekst kan dan later ingevoegd worden.
T- Met T wordt de tekst @ en de cursor van het scherm verwijderd en in de correctiebuffer opgeslagen voor later gebruik.
R- Als op R gedrukt wordt verschijnt op de onderste regel R: Hierachter ingetikte tekst vervangt, na het drukken van COPY, de tekst tussen @ en de cursor en wordt bovendien in de correctiebuffer opgeslagen.
X- Met X wordt het letterteken dat door de cursor aangegeven wordt vervangen door het letterteken dat na X getikt wordt. Als na X op COPY gedrukt wordt, dan wordt ALLES na de cursor gewist.
I - Met I wordt het letterteken dat na I getikt wordt ingevoegd VOOR de cursor.

ZOEKEN EN VERVANGEN:
1. Zoeken van tekst:
F - Met het commando F kan de tekst op het scherm afgezocht worden naar het voorkomen van een opgegeven stuk tekst (of een woord of letterteken).
Vanaf het begin van de huidige schermpagina worden alle plaatsen opgezocht waar de opgegeven tekst voorkomt. Deze kan dan selectief of op alle voorkomende plaatsen weggelaten worden of vervangen door een andere tekst.
De te vinden tekst moet omsloten worden door een teken dat niet in de te doorzoeken tekst voorkomt. (Bijv. / of ! of &.) Het zoekcommando gaat er dan bijvoorbeeld als volgt uitzien: F/huis/
Als hierna op COPY gedrukt wordt verschijnt het tekstgedeelte dat begint met het gezochte stuk tekst (waar het het EERST voorkomt) linksboven op het scherm.
2. Zoeken en vervangen:
Als na het te vinden stuk tekst een tweede stuk tekst opgegeven wordt, dat afgesloten wordt met hetzelfde teken, dan kan dit het eerste stuk tekst vervangen. Het F-commando gaat er dan bijv. als volgt uitzien:
/huis/woning/
Als er nu op COPY gedrukt wordt, dan verschijnt de tekst die begint met het eerste voorkomen van "huis" linksboven op het scherm.
Beneden op het scherm verschijnt
F:

Dit gebeurt iedere keer dat het woord "huis" gevonden wordt.
Als hierna Y wordt getikt, wordt op die plaats "huis" vervangen door "woning". Als N wordt getikt, wordt er niet vervangen. Dit gaat door tot alle plaatsen waar "huis" stond geweest zijn.
Met ESC kan het F-commando tussentijds uitgeschakeld worden.
Als het F-commando afgesloten wordt met A of a dan wordt AUTOMATISCH het eerste stuk tekst op ALLE plaatsen vervangen door het tweede. Bijv.:
F/huis/woning/a vervangt "huis" overal door "woning".
Als het tweede tekstgedeelte "leeg" is, dan kan de te vinden tekst worden gewist. Met bijv.: F/huis// kan "huis" overal gewist worden. Met: F/huis//A overal in 1 keer.

WEGSCHRIJVEN OF LADEN VAN "FILES":
Voor het wegschrijven van teksten (files) naar disk of tape wordt het commando > gebruikt.
Na > kan een naam opgegeven worden, bijv.:
> BRIEF 8.8.82
Als nu op RETURN gedrukt wordt, dan wordt de gehele tekst naar disk of tape geschreven.
Voor het laden van een tekst van disk of tape wordt het commando < gebruikt. Bijv.:
< NOTULEN
Als hierna op RETURN gedrukt wordt dan wordt het gehele file van de disk of tape geladen. Als er met cassette gewerkt wordt dan geeft het Cassette Operating System (COS), nadat op RETURN gedrukt is, het bericht PLAY TAPE of RECORD TAPE. Hierna moet de cassetterecorder aangezet worden, waarna op de spatiebalk gedrukt moet worden. Nu wordt de tekst geladen of weggeschreven.

DIVERSE COMMANDO'S:
W - Met W kan worden nagegaan hoever het tekstgeheugen (het lage geheugen van de ATOM) gevuld is. Na het intikken van W verschijnt het hexadecimale adres van het einde van de tekst op het scherm. Het tekst geheugen begint op #2800 en eindigt in een volledig intern uitgebreide ATOM op #3C00. Met externe geheugenkaarten kan het tekstgeheugen uitgebreid worden tot #8000.
* - Achter * kan een COS of DOS commando gebruikt worden vanuit de editor.
Bijv. met: *CAT
kan een catalogus opgevraagd worden van alle files op een tape of disk. Na het gebruiken van een * commando moet op de spatiebalk gedrukt worden om naar het tekstverwerkingsprogramma terug te keren.
Het toevoegen van tekst van tape of disk aan een al geladen tekst gaat als volgt: Vraag met W het geheugenadres op van de al geladen tekst en laad de toe te voegen tekst op dat adres met *LOAD. Als bijvoorbeeld na het intikken van W het adres 29AE verschijnt kan het file "deel twee" met:
*LOAD "deel twee" 29AE toegevoegd worden.
Q - Met Q wordt de editor verlaten, de tekstspace-pointer wordt op #82 gezet en de inhoud van het tekstgeheugen wordt in het hoge geheugen (vanaf #8200) van de ATOM geladen, alsof deze vanaf het toetsenbord was ingetikt. Zo wordt de tekst van een BASIC-programma dat met de tekstverwerker is ingetikt automatisch omgezet in een werkend programma in het hoge geheugen. De inhoud van het tekstgeheugen van de editor blijft ongewijzigd, zodat door EDIT te tikken de tekstverwerker opnieuw aangezet kan worden en het programma eventueel nog verder gewijzigd kan worden.
O - Met O wordt de tekst bewerkt en naar de printer gestuurd en afgedrukt. Daarvoor moeten eerst bewerkings-commando's in de tekst ingevoegd zijn. Deze commando's worden verderop behandeld.

FOUTMELDINGEN:

Pieptoon - Geeft aan dat de cursor op een ongeldige plaats staat, bijv. voorbij het einde van de tekst. Een poging om dan commando's , A, B, of R te gebruiken geeft een pieptoon. Een pieptoon kan ook betekenen dat ← twee keer tegelijk gebruikt wordt om een plaats in de tekst te markeren. Als dit gebeurt moet op ESC gedrukt worden om de fout op te heffen.

** Press COPY - Als deze melding verschijnt terwijl een van de commando's A, B, R, of T gebruikt is, dan betekend dit dat de correctiebuffer vol is. Door op COPY te drukken wordt teruggekeerd naar de editor en de inhoud van de correctiebuffer bewaard. Deze moet nu eerst in de tekst ingevoegd worden voor doorgegaan kan worden met intikken van tekst.

Als de melding verschijnt terwijl het commando E gebruikt wordt om tekst in te tikken, dan is het tekstgeheugen van de editor vol. De enige oplossing is het wegschrijven van de tekst naar tape of disk en met een nieuw file te beginnen.

** Full - Deze melding geeft aan dat tekst die ingetikt is achter A of B te lang is om nog in het tekstgeheugen erbij te passen. De inhoud van de correctiebuffer blijft bewaard maar wordt niet in de tekst ingevoegd. De inhoud van het tekstgeheugen dient op tape of disk weggeschreven te worden en de inhoud van de correctiebuffer overgebracht naar het tekstgeheugen.

** Press SPACE - Iedere poging om tekst weg te halen voorbij het begin van de tekst die achter A, B, E of R ingetikt wordt, laat dit commando vervallen en geeft deze melding. Met een spatie kan weer naar de editor worden teruggekeerd.

Het intikken van niet bestaande commando's en het aanslaan van verkeerde toetsen wordt door het tekstverwerkingsprogramma genegeerd!

BEWERKINGSCOMMANDO'S:

De bewerkingscommando's (processing commands) moeten voor het afdrukken in de tekst worden ingevoegd om de vorm waarin de tekst wordt afgedrukt te bepalen. Ze moeten aan het begin van een zin staan; dus na de RETURN van de vorige zin, of aan het begin van het document.

De commando's bestaan uit een controleteken, dit is in normale gevallen een punt, gevolgd door een enkele kleine letter, en soms gevolgd door een getal. De volgende commando's kunnen gebruikt worden:

.1 - Eerste zin.

Dit commando MOET gebruikt worden om het begin van een document aan te geven. Het initialiseert alle bewerkingscommando's op hun standaardwaarde (default setting), d.w.z. de waarden die alle commando's automatisch aannemen tenzij een ander waarde opgegeven wordt.

.l en .r - Paginagrootte.

Het commando .l dient ook om het aantal regels tekst per pagina te bepalen. Dit kan door achter .l het gewenste aantal te tikken, bijv: .135

De standaardwaarde is 58 regels per pagina.

Het commando .r dient om de regellengte en dus de plaats van de rechtermarge te bepalen. De standaardwaarde is 70 tekens.

.p .e en .w - Paginanummering.

Als de bladzijden van een document genummerd moeten worden dan kan met .p het gewenste nummer van de beginpagina van het document opgegeven worden. Met .p0 kan de bladzijdenummering weer afgezet worden. De standaard is geen bladzijdenummering. De nummering staat standaard gelijk met de rechtermarge op de oneven en met de linkermarge op de even pagina's.

Met .w wordt bepaald hoever de nummering uit het midden staat. Met de standaardwaarde 70 komt de nummering gelijk met de linker en rechtermarge.

Als .w een kleinere waarde heeft dan komt de nummering dichter naar het midden te staan.

Met .e wordt de nummering midden onder de pagina's gezet.

.j en .n - Uitvullen tekst.
De tekst wordt standaard bij het afdrukken uitgevuld met extra spaties, waardoor de rechtermarge een rechte lijn wordt. Dit uitvullen kan worden afgezet met .n en weer aangezet met .j.
.s en .d - Spatiering.
De tekst wordt standaard afgedrukt met enkele spatiering. Met .d kan op dubbele spatiering overgegaan worden en met .s weer op enkele.
.i .t en .m - Inspringen.
Een stuk tekst kan men laten inspringen met het commando .i gevolgd door het aantal posities dat de linker marge moet opschuiven. Bijv. :

> .i7Deze tekst wordt 7 spaties naar rechts afgedrukt. Dit blijft gehandhaafd tot het inspringen is afgezet met het commando .i zonder getal.

Met .t plus een getal wordt een enkele regel ingesprongen, onafhankelijk van een eventueel .i commando. Het .t commando kan dus gebruikt worden om in te springen aan het begin van een nieuwe paragraaf.
Het .m commando staat gelijk aan .t0 , maar het negeert een getal er achter. Het kan dus gebruikt worden om voor een met .i ingestelde marge nummers af te drukken.
.c - Centreren tekst.
Met .c wordt tekst die smaller is dan de regellengte gecentreerd. Als er marges verschoven zijn, dan wordt de tekst tussen de marges gecentreerd.
.a - Ruimte maken.
Het .aN commando geeft aan dat de volgende N regels op dezelfde bladzijde moeten komen. Als er niet genoeg ruimte is tot onderaan de bladzijde dan wordt op een nieuwe bladzijde begonnen vanaf het punt waar .aN in de tekst voorkomt. Bijv. met: .a4 Lijst van commando's:
wordt er voor gezorgd dat tenminste 3 regels van de lijst onder de kop op dezelfde pagina afgedrukt worden. Een lege ruimte voor afbeeldingen in de tekst kan gemaakt worden met bijv. .a20 gevolgd door 20 lege regels (20 x RETURN). Een nieuwe bladzijde kan worden begonnen met .a zonder getal.
.k en .o - Besturing printer.
De printer drukt standaard de gehele tekst achter elkaar door af, voor gebruik van kettingpapier. Als met losse vellen gewerkt wordt moet .k gebruikt worden. Daardoor wordt per pagina afgedrukt, waarna het bericht:
Paper
op het scherm verschijnt. Na het inzetten van een nieuw vel kan op een willekeurige toets gedrukt worden, waarna een volgende pagina afgedrukt wordt, enz.
Met het .oN commando kunnen speciale "control characters" naar de printer gestuurd worden. Hierbij moet voor N de decimale code van het character ingevuld worden. Deze control characters kunnen per printer verschillen.
./ - Commentaar.
Een zin die begint met ./ wordt genegeerd bij het afdrukken en kan gebruikt worden om commentaar aan een tekst toe te voegen. Een andere mogelijkheid is het niet laten afdrukken van een stuk tekst zonder dit te wissen.
.x - Veranderen controle tekens.
Met .x kan het controleteken voor bewerkingscommando's dat normaal een punt is, veranderd worden in een willekeurig ander teken. Bijv.: .x& verandert het controleteken van . in & en &x. verandert het dan weer terug in .
Aan het begin van een nieuw "file" is het controleteken steeds een punt.

BEWERKING VAN LANGE DOCUMENTEN.
Documenten die te lang zijn om in het tekstgeheugen van de ATOM te passen dienen in "files" van een geschikte lengte onderverdeeld te worden, die apart op tape of disk bewaard worden. De correctiebuffer wordt niet beinvloed door het laden of wegschrijven van tekst en kan dus gebruikt worden om tekst uit het ene "file" in het andere over te brengen. Om het hele document af te drukken kunnen de "files" na elkaar geladen, bewerkt en afgedrukt wordem. Hierbij blijven de bewerkingscommando's die aan het begin van het EERSTE file gebruikt zijn geldig,tenzij ze veranderd worden.

BEWERKEN VAN BASIC PROGRAMMA'S.
Het TEXT commando.
Om BASIC programma's te kunnen bewerken dienen deze eerst in de goede vorm omgezet te worden. Dit gaat als volgt in zijn werk: Laad het programma, in dit voorbeeld PROG genaamd, in het hoge tekstgeheugen door de volgende commando's in te tikken:

```
?18=#82
NEW
LOAD"PROG"
```

Hierna dient het TEXT commando aangeroepen te worden door TEXT te tikken.
Als er geen Floating Point ROM in de ATOM zit dient het commando aangeroepen te worden met LINK #ACFA.
Het commando TEXT zorgt ervoor dat alles wat op het scherm afgedrukt wordt in het tekstgeheugen terechtkomt, alsof het was ingetikt.
Als dus na het aanroepen van TEXT het programma "gelist" wordt met het commando LIST, dan komt het programma in de juiste vorm in het tekstgeheugen terecht. Nu kan met EDIT de tekstverwerker aangezet worden. Op de vraag Old Text? dient nu met Y geantwoord te worden. Hierna kan het programma bewerkt worden. Tenslotte kan het gewijzigde BASIC programma met het commando Q weer in het hoge geheugen van de ATOM geladen worden en geprobeerd worden.
BELANGRIJK: Hierbij worden geen programmaregels van meer dan 64 charcters lengte geaccepteerd.

Het aanroepen van de tekstverwerker vanuit BASIC.
De tekstverwerker kan worden aangeroepen vanuit een BASIC programma in het hoge geheugen van de ATOM door in het programma LINK #ACF7 uit te voeren.
Op deze manier kunnen "text files" gemaakt worden of gewijzigd door een BASIC programma met gebruikmaking van de faciliteiten van de tekstverwerker.



NAGEKOMEN BERICHT:
Bij het bandjes archief zijn momenteel in voor bereiding:
Bandje 15 met programma's uit de jaargang 1983.
Bandje 16 met diverse databases.

OVERZICHT COMMANDO'S.

Invoercommando's:

toets	wijzigt tekst	wijzigt buff	beschrijving
A	x	x	invoegen na
B	x	x	invoegen voor
C		x	kopie tekst naar buffer
COPY			einde tekst
DELETE	x		wissen
E	x		intikken tekst
ESC			wissen markering
F			zoeken (en vervangen)
H			cursor linksboven
I			invoegen teken
N			volgende schermpagina
O			afdrukken op printer
P			vorige schermpagina
Q			naar BASIC programma
R	x	x	vervangen tekst
S			naar begin tekst
T		x	tekst naar buffer
W			adres einde tekst
X	x		wijziging teken
Z			naar einde tekst
↑			tekst een regel omhoog
[			cursor naar begin regel
]			cursor naar einde regel
@			markeren cursorpositie
<			laden "tekstfile"
>			wegschrijven "tekstfile"
*			uitvoeren COS/DOS commando

Bewerkings commando's

- .a nieuwe bladzijde
- .c centreren regel
- .d dubbele spatiering in volgende regels
- .e paginanr.'s in het midden
- .iN inspringen met N spaties
- .j uitvullen volgende regels
- .k per pagina afdrukken
- .lN N regels per pagina
- .l eerste regel document
- .m marge gelijk aan .tO
- .n niet uitvullen volgende regels
- .oN control character N naar printer
- .pN paginering begint met nummer N
- .pO geen paginering
- .rN regellengte naar N tekens
- .s enkele spatiering volgende regels
- .tN inspringen 1 regel met N spaties
- .w breedte paginanummers
- .x? verandert controleteken in teken ?
- ./ commentaar - wordt niet afgedrukt

VRAAG:

Henk Reinders heeft de basic-interpreter, floating-point en JOS-box in RAM-geheugen. Bij het statement GRMOD worden in de interpreter 20 bytes veranderd en wel de adressen C200,C220 enz. De verandering bestaat uit het inverteren van bit 7. Wie weet het antwoordt op dit raadsel.

Nu er eindelijk schot in BIG BENNY lijkt te zitten is het tijd geworden voor enige aanvullende informatie. We zullen globaal ingaan op de aansturing van dit klokje aan de hand van enkele programma's. Ook zullen we laten zien hoe er met behulp van BIG BENNY interrupts gegenereerd kunnen worden van 1 sec, 1 min en 1 uur.

D E A A N S T U R I N G

Aangezien het gebruikte klok IC niet rechtstreeks door de microprocessor kan worden aangestuurd, maken we in dit ontwerp gebruik van een PIA. Deze PIA - de 6821 - heeft twee 8-bits poorten en 4 control-lijnen, die software-matig gestuurd kunnen worden.
De PIA bestaat intern uit 6 registers verdeeld over 4 adressen: 2 Peripheral Registers, 2 Data Direction Registers en 2 Control Registers. In de volgende tabel staat aangegeven hoe we toegang krijgen tot de verschillende registers.

adres	control register bit CRA-2	control register bit CRB-2	location selected
B400	1	X	Peripheral Register A
B400	0	X	Data Direction Register A
B401	X	X	Control Register A
B402	X	1	Peripheral Register B
B402	X	0	Data Direction Register B
B403	X	X	Control Register B

X=Don't care (doet er niet toe)

Bitje 2 in de Respectievelijke Control Registers bepaalt of we toegang hebben tot het Peripheral Register of het Data Direction Register (DDR). Elk hoge bitje in het DDR selecteert de overeenkomstige bit in het Peripheral Register als uitgang, elk lage bit staat voor een ingang.

De verschillende bitjes in een Control Register besturen de Control- en Interruptsignalen als volgt:

bit 7	6	5 4 3	2	1 0
IRQA1	IRQA2	CA2-Control	DDR-A	CA1-Control
			Access	

Voor de verdere gegevens van dit IC verwijzen wij naar de Datasheet (drukwerkarchief ?).

De PIA-poorten worden gebruikt om te communiceren met het klok-IC. De interruptlijnen zijn via LK 3 in onze computer met de 6502. Alle aansluitingen zijn weergegeven in het fraaie schema, dat getekend werd door Rudi van Drunen.

```
 10 REM BIG-BENNY IN BASIC
 20 REM CLOCK-PROGRAMMA
 30 GOSUB i;REM INIT
 40 PRINT $12;FOR I=1 TO 30;WAIT;NEXT
 50 IF A=0 GOTO 70
 60 GOSUB s;REM SET TIME
 70 PRINT $12$8;?#E1=0
 80 DO
 90 GOSUB t
100 ?#DE=#C0;PRINT $T'
110 UNTIL 0
120sPRINT $12'''
130 INPUT "UREN "U
140 INPUT "MIN "M
150 B?3=#34;REM CB2 low --> HOLD = high & B402=PRB
160 X=5;Y=U/10+8;GOSUB w
170 X=4;Y=U%10;GOSUB w
180 X=3;Y=M/10;GOSUB w
190 X=2;Y=M%10;GOSUB w
200 X=1;Y=0;GOSUB w
210 X=0;Y=0;GOSUB w
220 B?3=#3C;REM CB2 high --> HOLD low & B402=PRB
230 RETURN
240 REM SET TIME IN $T
250 REM uses X as clock address, data return in D
260tB?3=#34;REM AS LINE G.150
270 X=5;GOSUB r;?T=D+48
280 X=4;GOSUB r;T?1=D+48
290 T?2=CH":"
300 X=3;GOSUB r;T?3=D+48
310 X=2;GOSUB r;T?4=D+48
320 T?5=CH":"
330 X=1;GOSUB r;T?6=D+48
340 X=0;GOSUB r;T?7=D+48
350 B?3=#3C;REM AS LINE G.220
360 T?8=#D
370 RETURN
380 REM INIT
390iB=#B400;T=#3300
400 B?3=0;REM clear CRB-2 to sel. DDRB at B402
410 B?2=#C0;REM set PRB-6&7 as outputs
420 B?3=#3C;REM B402=PRB /CB2 as high output /disable interr.
430 B?2=0;REM all B-outputs low
440 RETURN
450 REM READ CLOCK
460rB?1=0;REM B400=DDRA
470 ?B=#F0;REM PA0-3 input(data) & PA4-7 output(clock-address)
480 B?1=4;REM B400=PRA
490 ?B=X*16;REM address on high nipple
500 B?2=#80;REM enable READ
510 D=?B&#0F;REM data on low nipple
520 B?2=0;REM disable READ
530 IF X=5;D=D&3
540 RETURN
550 REM WRITE IN CLOCK
560wB?1=0;REM B400=DDRA
570 ?B=#FF;REM all PA's as outputs
580 B?1=4;REM B400=PRA
590 ?B=X*16+Y;REM B400:high nipple=clock address/ low=data
600 B?2=#40;REM enable WRITE (on PB6)
610 B?2=0;REM disable WRITE
620 RETURN
```

FUNCTION TABLE FIGURE 1

ADDRESS INPUTS				INTERNAL COUNTER	DATA I/O				DATA LIMITS	NOTES
A_0	A_1	A_2	A_3		D_0	D_1	D_2	D_3		
0	0	0	0	S 1	*	*	*	*	0∿9	S1 or S10 are reset to zero irrespective of input data D0∿D3 when write instruction is executed with address selection.
1	0	0	0	S 10	*	*	*		0∿5	
0	1	0	0	MI 1	*	*	*	*	0∿9	
1	1	0	0	MI 10	*	*	*		0∿5	
0	0	1	0	H 1	*	*	*	*	0∿9	
1	0	1	0	H 10	*	*	†	†	0∿1 / 0∿2	D2 = "1" for PM D3 = "1" for 24 hour format D2 = "0" for AM D3 = "0" for 12 hour format
0	1	1	0	W	*	*	*		0∿6	
1	1	1	0	D 1	*	*	*	*	0∿9	
0	0	0	1	D 10	*	*	†		0∿3	D2 = "1" for 29 days in month 2 (2) D2 = "0" for 28 days in month 2
1	0	0	1	MO 1	*	*	*	*	0∿9	
0	1	0	1	MO 10	*				0∿1	
1	1	0	1	Y 1	*	*	*	*	0∿9	
0	0	1	1	Y 10	*	*	*	*	0∿9	

PIN CONFIGURATION

Pin	Name	Pin	Name
1	VCC	18	HOLD
2	WRITE	17	XT
3	READ	16	XT
4	A_0	15	·30 ADJ
5	A_1	14	TEST
6	A_2	13	GND
7	A_3	12	D_3
8	CS	11	D_2
9	D_0	10	D_1

(1) * data valid as "0" or "1".

Blank does not exist (unrecognized during a write and held at "0" during a read)

† data bits used for AM/PM, 12/24 HOUR and leap year.

(2) If D2 previously set to "1", upon completion of month 2 day 29, D2 will be internally reset to "0".

D E I N T E R R U P T

Ter illustratie van de verschillende interrupts het volgende machine-taal programma:

```
     \initialiseer
     \1 sec - interval
:KK0 SEI         zet interrupt uit
     LDA @#05  zet CB1-interrupt en DDR-bit
     STA #B403      in het Control Register
     LDA #B402 lees data (reset interrupt)
     CLI         sta interrupt toe
     RTS

     \interrupt-routine
:KK1 TXA;PHA   bewaar X-reg op stack
     TYA;PHA   bewaar Y-reg op stack
     JSR #FD1A belletje !!! (bv.)
     PLA;TAY   haal Y-reg van stack
     PLA;TAX   haal X-reg van stack
     LDA #B402 lees data (reset interrupt)
     PLA         haal accumulator van stack
     RTI         return from interrupt
```

Uiteraard moet nog wel de interrupt-vector worden geprogrammeerd:

```
  ?#204=KK1%256
  ?#205=KK1/256
```

Vervolgens kan de routine worden gestart met: LINK KK0

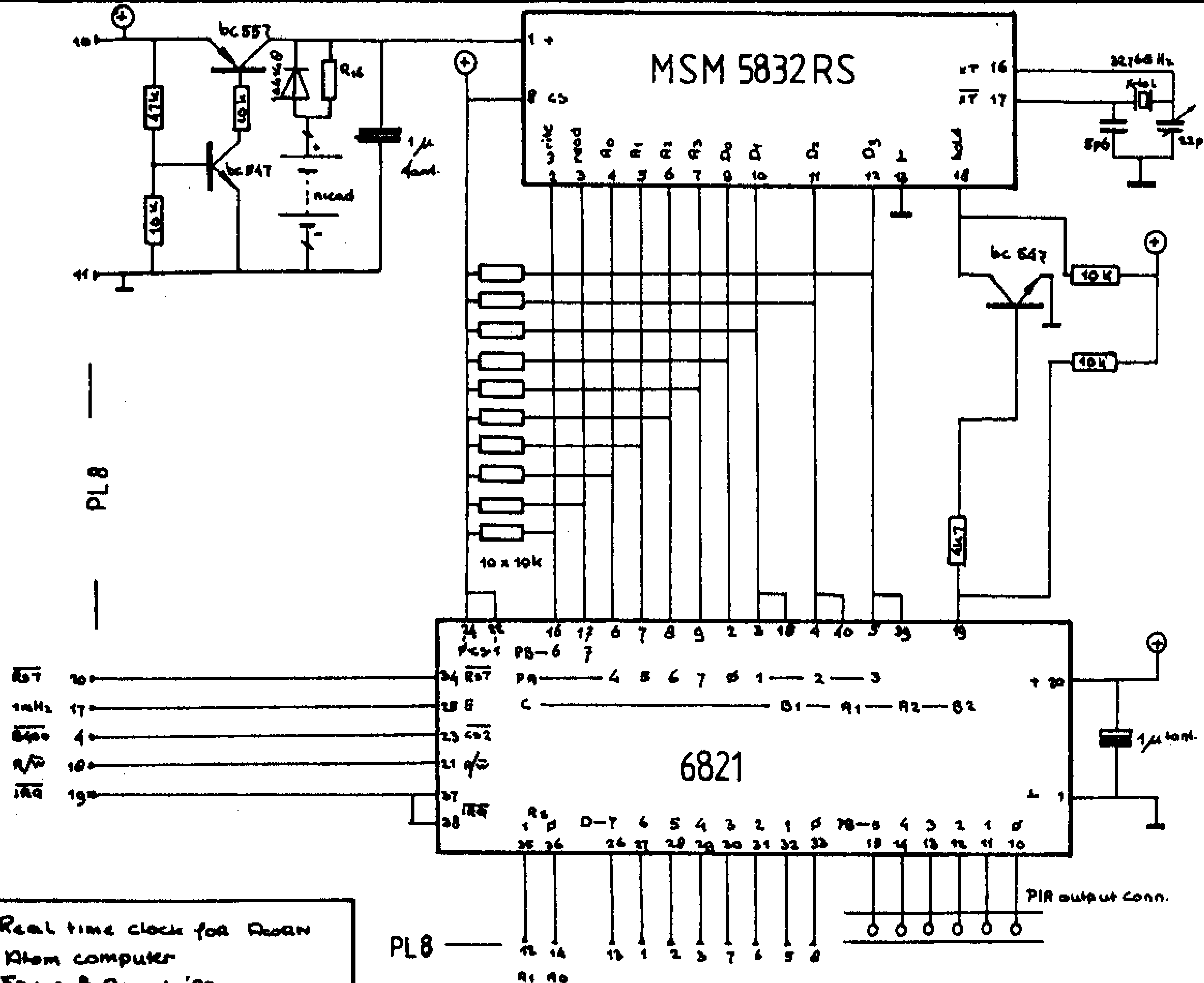
Ook mogelijk zijn intervallen van 1 minuut en 1 uur.

```
     \1 min            \1 uur
:KK0 SEI               SEI
     LDA @#05          LDA @#1C
     STA #B401         STA #B401
     LDA #B400         LDA #B400
     CLI               CLI
     RTS               RTS

:KK1 TXA;PHA           TXA;PHA
     TYA;PHA           TYA;PHA
     JSR #FD1A         JSR #FD1A
     PLA;TAY           PLA;TAY
     PLA;TAX           PLA;TAX
     LDA #B400         LDA #B400
     PLA               PLA
     RTI               RTI
```

Veel succes
RUURD & FRANS

PS. in samenwerking met de VIA is het zeer goed mogelijk om bv. een klok te maken die op 1/100 sec nauwkeurig loopt. In dat geval kan de 1 sec interrupt van BIG BENNY gebruikt worden om de VIA te synchroniseren.


MSM 5832RS
6821
bc557
bc547
47k
10k
R16
nicad
1µ tant.
Xtal
5p6
22p
10 x 10k
10 K
4k7
PL8
RST
1MHz
R/W
IRQ
PIR output conn.
Real time clock for Acorn
Atom computer
Frans & Ruurd '83
get. Rudi v.Drunen 191283

```
  10REM SOMMEN TEST
 100DIMB8,I(-1);S=0;T=60
 110GOS.a;!#80=-(T*60)
 120$B="GOTO 700";?16=B;?17=B/256
 130P.$12"JE KRIJGT 10 SOMMEM"'
 140@=0;P."TE MAKEN IN "T" SEKONDEN"''
 200F.Q=1TO10;P."SOM "Q" IS :"''
 300GOS.(1000+A.R.%4*100)
 400IN.$I;IF$I="" P.$7;G.400
 410Z=0
 420F.P=0TO LEN(I)-1
 430IFI?P(CH"0"ORI?P)CH"9";P.$7;G.400
 440Z=I?P-CH"0"+10*Z
 450N.
 500IFZ=R P.'"GOED !";S=S+1;G.520
 510P.'"SORRY, HET MOEST ZIJN "R
 520F.A=1TO60;WAIT;N.
 530P.$12;@=0
 540N.
 600P."JE HEBT VAN DE 10 SOMMEN"'"ER "S" GOED"'
 610E=T+(!#80)/60;P."IN "E" SEKONDEN"''
 620IFS)8 P."GEFELICITEERD"'
 630IFS*T/E)10 P."GOED ZO"'
 640?#20A=?#A0;?#20B=?#A1;E.
 700P.&7$12;@=0;P."* DE TIJD IS VOORBIJ *"'
 710!#80=0;G.600
1000X=2+A.R.%98;Y=2+A.R.%98
1010R=X+Y;P."WAT IS "X" + "Y" = ";R.
1100X=2+A.R.%98;Y=1+A.R.%98
1110IFY+2)X G.1100
1120R=X-Y;P."WAT IS "X" - "Y" = ";R.
1200X=2+A.R.%11;Y=2+A.R.%11
1210R=X*Y;P."WAT IS "X" * "Y" = ";R.
1300X=2+A.R.%11;Y=2+A.R.%11
1310Y=X*R;P."WAT IS "Y" : "X" = ";R.
9000aP.$21;P=#84
9010↑;PHP;CLD;STX#E4;STY#E5
9020BIT#B002;BVC#94;JSR#21C;BCC#8A
9030JSR#FB8A;JSR#21C;BCS#97;JMP#FEA7;←
9040P=#21C
9050↑;JSR#FE66;INC#80;BNE#228
9060INC#81;BNE#228;BRK;JMP#FE71;←
9070?#A0=?#20A;?#A1=?#20B
9080?#20A=#E4;?#20B=0;P.$6;R.
```

Het overkomt mij weleens dt ik m.b.v. het text statement uit de tekstverwerker uitvoer van een andere toolkit op #Axxx (b.v. de JOS-box) wil opslaan. Dit is dan onmogelijk omdat beide toolkits op #Axxx zitten.
Onderstaande routine is een verbeterde versie van het text-statement. Als U naar het start adres LINKT (U kunt daarvoor natuurlijk ook een nieuw statement maken) wordt alles dat wordt afgdrukt opgeslagen vanaf #2800. Door de routine in niet schakelbaar geheugen te zetten kunt u nu alle uitvoer opslaan.

```
 10 REM TEXT STATEMENT
 20 REM RONALD BOERS
 25 IN."START ASSEMBL."O;REM STARTADRES
 30 E=#70
 40 W=#28;REM OPSLAG GEBIED IS #2800
 50 K=O;@=0
 60 DIM LL5;FOR L=0 TO 5;LLL=-1;N.
 70 P.$21;FOR Q=1 TO 2;P=O;[
 75\ INIT. ROUTINE AL EENS GELINKT ?
 80:LL0LDA#208;CMP@LL1&#FF;BNELL5
 90LDA#209;CMP@LL1/256;BEQLL4
120:LL5LDA#208;STA K
130LDA #209;STA K+1
140 LDA@LL1&#FF;STA #208
150 LDA@LL1/256;STA #209
160:LL4LDA@0;STA E
170 LDA@W;STA E+1
180 RTS
190:LL1 \HOOFD ROUTINE
200 STY E+2;PHA
210 CMP@#0D;BEQ LL2
230 CMP@#20;BCC LL3
240:LL2 LDY@00;STA (E),Y
250 INY;LDA@#04;STA (E),Y
260 INC E;BNE LL3;INC E+1
270:LL3 LDY E+2;PLA;JMP (K)
280]K=P;N.;P.$6
290 P.'"VAN "&O" TOT "&P+1'
300 END
```

OVERGENOMEN UIT DATA-CHECK.

```
 10 REM KNIPPERROUTINE ACORN NIEUWS 4 BLZ 32
 20 REM VERBETERD DOOR AREN SLOOTWEG
 30      DIM LL2
 40      P=#2800
 50[
 60:LL1 LDY@0
 70      JSR#FB81    ;\WACHT 1/2 SEC
 80:LL2 LDA#8020,Y ;\WACHT ADRES KNIPPEREND VLAK.
 90      EOR@#80
100      STA#8020,Y ;\START ADRES KNIPPEREND VLAK
110      INY
120      CPY@#20     ;\LENGTE KNIPPEREND VLAK
130      BNELL2
140      LDA#B001
150      AND@#40     ;\CNTR TOETS INGEDRUKT.
160      BNELL1
170      RTS
180]
190      LINK#2800
200      END
```

DE VOLGENDE ROUTINE IS VAN EEN ONBEKENDE AUTEUR, IS OVERGENOMEN UIT BRONS GROEN EIKELTJE NR.1 1984.

```
 10 REM FLASH
 20 J=13;DIM LLJ;F.I=0 TO J;LLI=#FFF;N.;DIMA4;@=0
 30P.$12"FLASH"
 40IN.''"GEEF STARTADRES B.V. #3C00"Z;IF?18=Z/256;G.40
 50IN."PRINTER J/N"$A
 60LL8=#B002;LL9=#C3C8;LL10=#F88A;LL11=#FD44;LL12=#FE71
 70LL13=#FEAC
 80P.$21;F.I=1TO2;P=Z;IFI=2;P.$6;IF?A=74P$2
 90[
100:LL0PHP;CLD;STX#E4;STY#E5
110:LL2BITLL8;BVCLL3;JSRLL12;BCCLL2
120:LL3JSRLL10;LDA@#00
130:LL4PHA;LDA@#00
140:LL5PHA;JSRLL12;PLA;BCCLL6;TAY
150DEY;TYA;BNELL5;LDY#E0;JSRLL11
160PLA;TAY;INY;TYA;SEC
170BCSLL4
180:LL6PLA;AND@#01;BEQLL7;LDY#E0;JSRLL11
190:LL7JMPLL13;JSRLL9;LDA#52;STA#AE;LDA#53
200STA#AF;RTS
210];N.;P.'"CODE VAN"&Z" TOT"&P'$3
220?#20A=Z%256;?#20B=Z/256
230END
240 UIT BRONSGROEN EIKELTJE
```

HOORT BIJ TOETERS EN BELLEN VORIG NUMMER.

Dit heeft eveneens tot gevolg dat bij mij de cursor niet meer verkleurt bij SAVE en LOAD. Het kleuren mag n.l. alleen in tekst-mode plaatsvinden, en om te weten of het scherm in graphics of tekst staat, kijk je normalerwijze naar 7#B000. Is dat nul dan zijn we in tekst-mode, is dat ongelijk aan nul dan hebben we te maken met graphics. De Josbox kijkt echter naar het high-byte van de RDCH-vector (#FE → tekst; anders → graphics). Als het low-byte werd beschouwd, kon je het probleem nog omzeilen door de nieuwe routine op #E794 te zetten of zo.
Het terugzetten naar originele waarden komt overigens veel vaker voor (Word Pack b.v.).

En dan het statement READ, een hoofdstuk apart. De methode voor het afhandelen van dit statement is even geraffineerd als ingrijpend. De inputbuffer (vanaf #100) wordt n.l. misbruikt.
Gesteld dat RESTORE de datapointer op de juiste plaats heeft gezet, gebeurt er bij READ het volgende:

1. de string vanaf „READ" (of vanaf „," als er al eerder iets ingelezen is) tot aan „," of „:" of <CR> wordt inclusief eventuele spaties in de inputbuffer (vanaf # 100) gekopieerd
2. daarachter komt een „="
3. daarachter wordt de string vanaf de datapointer tot aan „," of „;" of <CR> gekopieerd, wederom inclusief eventuele spaties
4. daarachter komt „;LINK#A219<CR>"
5. het aldus ontstane assignment statement wordt geïnterpreteerd waarna LINK #A219 op zoek gaat naar het volgende READ-item.

Deze slimme handelswijze heeft tot gevolg dat in READ alles mag staan wat links van een =-teken kan voorkomen (A, TT2, 7#80, %F, !#8200, $C, LETV enz.) en in DATA overeenkomstige integer, real- en string-expressies. Het is nu ook duidelijk waarom strings tussen quotes (" ") moeten staan en waarom de lengte ervan beperkt is. Maken we deze nl. te lang dan gaat de string-buffer (vanaf #140) de inputbuffer overschrijven. Wat wel kan, gezien het voorgaande is dit:

```
RESTORE
DIM A80
READ $A, $A+LENA
DATA "DIT IS EEN STRING VAN 34 KARAKTERS"
DATA ": SAMEN MET NOG EEN 40 WORDEN DAT ER 74."
P.$A,LENA
END
```

De „," en „;" mogen niet in strings voorkomen om je erop te attenderen dat je de sluit-quotes vergeten bent. Ikzelf had liever gezien dat in ieder geval de komma wel in strings mocht voorkomen.
Ook de logische NOT is mogelijk:

```
READ B          gelijkwaardig met:
DATA (B=0)      IF B=0 THEN B=1 ELSE B=0
```

en als we het helemaal onleesbaar willen maken:

```
READ B=(B
DATA 0)
```